Chapter 17
The CALIS Procedure

Chapter Table of Contents

OVERVIEW .................................................. 397
   Structural Equation Models .................................. 400

GETTING STARTED ........................................ 407

SYNTAX .................................................. 414
   PROC CALIS Statement ..................................... 415
   COSAN Model Statement .................................... 437
   MATRIX Statement .......................................... 439
   RAM Model Statement ....................................... 443
   LINEQS Model Statement ................................... 447
   STD Statement ............................................. 450
   COV Statement ............................................. 451
   FACTOR Model Statement .................................. 453
   BOUNDS Statement .......................................... 454
   LINCON Statement .......................................... 455
   NLINCON Statement ......................................... 456
   NLOPTIONS Statement ...................................... 457
   PARAMETERS Statement .................................... 470
   STRUCTEQ Statement ....................................... 470
   VARNAMES Statement ...................................... 471
   BY Statement ................................................ 472
   VAR Statement ............................................... 472
   PARTIAL Statement ......................................... 473
   FREQ Statement ............................................ 473
   WEIGHT Statement .......................................... 473
   SAS Program Statements .................................... 474

DETAILS .................................................. 476
   Input Data Sets ............................................. 476
   Output Data Sets ........................................... 480
   Missing Values ............................................. 490
   Estimation Criteria ......................................... 490
   Relationships among Estimation Criteria .................... 493
   Testing Rank Deficiency ................................... 493
   Approximate Standard Errors ............................... 494
Chapter 17. The CALIS Procedure

Assessment of Fit ................................ 495
Measures of Multivariate Kurtosis ..................... 503
Initial Estimates .................................... 506
Automatic Variable Selection .......................... 507
Exogenous Manifest Variables ........................... 508
Use of Optimization Techniques ........................ 510
Modification Indices .................................. 519
Constrained Estimation Using Program Code ............ 520
Counting the Degrees of Freedom ........................ 522
Computational Problems ................................ 523
Displayed Output ...................................... 526
ODS Table Names ..................................... 533

EXAMPLES ............................................ 535
Example 17.1 Path Analysis: Stability of Alienation ........ 536
Example 17.2 Simultaneous Equations with Intercept ........ 556
Example 17.3 Second-Order Confirmatory Factor Analysis ....... 563
Example 17.4 Linear Relations Among Factor Loadings .......... 571
Example 17.5 Ordinal Relations Among Factor Loadings .......... 577
Example 17.6 Longitudinal Factor Analysis .................. 583

REFERENCES ......................................... 585
Chapter 17
The CALIS Procedure

Overview

Structural equation modeling using covariance analysis is an important statistical tool in economics and behavioral sciences. Structural equations express relationships among several variables that can be either directly observed variables (manifest variables) or unobserved hypothetical variables (latent variables). For an introduction to latent variable models, refer to Loehlin (1987), Bollen (1989b), Everitt (1984), or Long (1983); and for manifest variables, refer to Fuller (1987).

In structural models, as opposed to functional models, all variables are taken to be random rather than having fixed levels. For maximum likelihood (default) and generalized least-squares estimation in PROC CALIS, the random variables are assumed to have an approximately multivariate normal distribution. Nonnormality, especially high kurtosis, can produce poor estimates and grossly incorrect standard errors and hypothesis tests, even in large samples. Consequently, the assumption of normality is much more important than in models with nonstochastic exogenous variables. You should remove outliers and consider transformations of nonnormal variables before using PROC CALIS with maximum likelihood (default) or generalized least-squares estimation. If the number of observations is sufficiently large, Browne’s asymptotically distribution-free (ADF) estimation method can be used.

You can use the CALIS procedure to estimate parameters and test hypotheses for constrained and unconstrained problems in

- multiple and multivariate linear regression
- linear measurement-error models
- path analysis and causal modeling
- simultaneous equation models with reciprocal causation
- exploratory and confirmatory factor analysis of any order
- canonical correlation
- a wide variety of other (non)linear latent variable models

The parameters are estimated using the criteria of

- unweighted least squares (ULS)
- generalized least squares (GLS, with optional weight matrix input)
- maximum likelihood (ML, for multivariate normal data)
- weighted least squares (WLS, ADF, with optional weight matrix input)
diagonally weighted least squares (DWLS, with optional weight matrix input)

The default weight matrix for generalized least-squares estimation is the sample covariance or correlation matrix. The default weight matrix for weighted least-squares estimation is an estimate of the asymptotic covariance matrix of the sample covariance or correlation matrix. In this case, weighted least-squares estimation is equivalent to Browne’s (1982, 1984) asymptotic distribution-free estimation. The default weight matrix for diagonally weighted least-squares estimation is an estimate of the asymptotic variances of the input sample covariance or correlation matrix. You can also use an input data set to specify the weight matrix in GLS, WLS, and DWLS estimation.

You can specify the model in several ways:

- You can do a constrained (confirmatory) first-order factor analysis or component analysis using the FACTOR statement.
- You can specify simple path models using an easily formulated list-type RAM statement similar to that originally developed by J. McArdle (McArdle and McDonald 1984).
- If you have a set of structural equations to describe the model, you can use an equation-type LINEQS statement similar to that originally developed by P. Bentler (1985).
- You can analyze a broad family of matrix models using COSAN and MATRIX statements that are similar to the COSAN program of R. McDonald and C. Fraser (McDonald 1978, 1980). It enables you to specify complex matrix models including nonlinear equation models and higher-order factor models.

You can specify linear and nonlinear equality and inequality constraints on the parameters with several different statements, depending on the type of input. Lagrange multiplier test indices are computed for simple constant and equality parameter constraints and for active boundary constraints. General equality and inequality constraints can be formulated using program statements. For more information, see the “SAS Program Statements” section on page 474.

PROC CALIS offers a variety of methods for the automatic generation of initial values for the optimization process:

- two-stage least-squares estimation
- instrumental variable factor analysis
- approximate factor analysis
- ordinary least-squares estimation
- McDonald’s (McDonald and Hartmann 1992) method

In many common applications, these initial values prevent computational problems and save computer time.

Because numerical problems can occur in the (non)linearly constrained optimization process, the CALIS procedure offers several optimization algorithms:
• Levenberg-Marquardt algorithm (Moré, 1978)
• trust region algorithm (Gay 1983)
• Newton-Raphson algorithm with line search
• ridge-stabilized Newton-Raphson algorithm
• various quasi-Newton and dual quasi-Newton algorithms: Broyden-Fletcher-Goldfarb-Shanno and Davidon-Fletcher-Powell, including a sequential quadratic programming algorithm for processing nonlinear equality and inequality constraints
• various conjugate gradient algorithms: automatic restart algorithm of Powell (1977), Fletcher-Reeves, Polak-Ribiere, and conjugate descent algorithm of Fletcher (1980)

The quasi-Newton and conjugate gradient algorithms can be modified by several line-search methods. All of the optimization techniques can impose simple boundary and general linear constraints on the parameters. Only the dual quasi-Newton algorithm is able to impose general nonlinear equality and inequality constraints.

The procedure creates an OUTRAM= output data set that completely describes the model (except for program statements) and also contains parameter estimates. This data set can be used as input for another execution of PROC CALIS. Small model changes can be made by editing this data set, so you can exploit the old parameter estimates as starting values in a subsequent analysis. An OUTEST= data set contains information on the optimal parameter estimates (parameter estimates, gradient, Hessian, projected Hessian and Hessian of Lagrange function for constrained optimization, the information matrix, and standard errors). The OUTEST= data set can be used as an INEST= data set to provide starting values and boundary and linear constraints for the parameters. An OUTSTAT= data set contains residuals and, for exploratory factor analysis, the rotated and unrotated factor loadings.

Automatic variable selection (using only those variables from the input data set that are used in the model specification) is performed in connection with the RAM and LINEQS input statements or when these models are recognized in an input model file. Also in these cases, the covariances of the exogenous manifest variables are recognized as given constants. With the COSAN, RAM, and LINEQS statements, you can display the predetermined pattern of constant and variable elements in the predicted model matrix before the minimization process starts. For more information, see the section “Automatic Variable Selection” on page 507 and the section “Exogenous Manifest Variables” on page 508.

PROC CALIS offers an analysis of linear dependencies in the information matrix (approximate Hessian matrix) that may be helpful in detecting unidentified models. You also can save the information matrix and the approximate covariance matrix of the parameter estimates (inverse of the information matrix), together with parameter estimates, gradient, and approximate standard errors, in an output data set for further analysis.

Release 6.11 of PROC CALIS does not provide the analysis of multiple samples with different sample size or a generalized algorithm for missing values in the data.
However, the analysis of multiple samples with equal sample size can be performed by the analysis of a moment supermatrix containing the individual moment matrices as block diagonal submatrices.

### Structural Equation Models

#### The Generalized COSAN Model

PROC CALIS can analyze matrix models of the form

\[
 C = F_1P_1F_1' + \cdots + F_mP_mF_m'
\]

where \( C \) is a symmetric correlation or covariance matrix, each matrix \( F_k, k = 1, \ldots, m \), is the product of \( n(k) \) matrices \( F_{k_1}, \ldots, F_{k_n(k)} \), and each matrix \( P_k \) is symmetric, that is,

\[
 F_k = F_{k_1} \cdots F_{k_n(k)} \quad \text{and} \quad P_k = P_k', \quad k = 1, \ldots, m
\]

The matrices \( F_{k_j} \) and \( P_k \) in the model are parameterized by the matrices \( G_{k_j} \) and \( Q_k \)

\[
 F_{k_j} = \begin{cases} 
 G_{k_j}^{-1} & j = 1, \ldots, n(k) \\
 (I - G_{k_j})^{-1} & \end{cases} \quad \text{and} \quad P_k = \begin{cases} 
 Q_k^{-1} 
\end{cases}
\]

where you can specify the type of matrix desired.

The matrices \( G_{k_j} \) and \( Q_k \) can contain

- constant values
- parameters to be estimated
- values computed from parameters via programming statements

The parameters can be summarized in a parameter vector \( X = (x_1, \ldots, x_t) \). For a given covariance or correlation matrix \( C \), PROC CALIS computes the unweighted least-squares (ULS), generalized least-squares (GLS), maximum likelihood (ML), weighted least-squares (WLS), or diagonally weighted least-squares (DWLS) estimates of the vector \( X \).

#### Some Special Cases of the Generalized COSAN Model

**Original COSAN (Covariance Structure Analysis) Model** (McDonald 1978, 1980)

Covariance Structure:

\[
 C = F_1 \cdots F_nPF_n' \cdots F_1'
\]
RAM (Reticular Action) Model (McArdle 1980; McArdle and McDonald 1984)

Structural Equation Model:

\[ v = A v + u \]

where \( A \) is a matrix of coefficients, and \( v \) and \( u \) are vectors of random variables. The variables in \( v \) and \( u \) can be manifest or latent variables. The endogenous variables corresponding to the components in \( v \) are expressed as a linear combination of the remaining variables and a residual component in \( u \) with covariance matrix \( P \).

Covariance Structure:

\[ C = J (I - A)^{-1} P ((I - A)^{-1})' J' \]

with selection matrix \( J \) and

\[ C = \mathcal{E}\{J vv'J'\} \quad \text{and} \quad P = \mathcal{E}\{uu'\} \]

LINEQS (Linear Equations) Model (Bentler and Weeks 1980)

Structural Equation Model:

\[ \eta = \beta \eta + \gamma \xi \]

where \( \beta \) and \( \gamma \) are coefficient matrices, and \( \eta \) and \( \xi \) are vectors of random variables. The components of \( \eta \) correspond to the endogenous variables; the components of \( \xi \) correspond to the exogenous variables and to error variables. The variables in \( \eta \) and \( \xi \) can be manifest or latent variables. The endogenous variables in \( \eta \) are expressed as a linear combination of the remaining endogenous variables, of the exogenous variables of \( \xi \), and of a residual component in \( \xi \). The coefficient matrix \( \beta \) describes the relationships among the endogenous variables of \( \eta \), and \( I - \beta \) should be nonsingular. The coefficient matrix \( \gamma \) describes the relationships between the endogenous variables of \( \eta \) and the exogenous and error variables of \( \xi \).

Covariance Structure:

\[ C = J (I - B)^{-1} \Gamma \Phi \Gamma^t ((I - B)^{-1})' J' \]

with selection matrix \( J \), \( \Phi = \mathcal{E}\{\xi \xi'\} \), and

\[ B = \begin{pmatrix} \beta & 0 \\ 0 & 0 \end{pmatrix} \quad \text{and} \quad \Gamma = \begin{pmatrix} \gamma \\ I \end{pmatrix} \]

Keesling - Wiley - Jöreskog LISREL (Linear Structural Relationship) Model

Structural Equation Model and Measurement Models:

\[ \eta = B \eta + \Gamma \xi + \zeta \quad , \quad y = \Lambda_y \eta + \epsilon \quad , \quad x = \Lambda_x \xi + \delta \]
Chapter 17. The CALIS Procedure

where \( \eta \) and \( \xi \) are vectors of latent variables (factors), and \( x \) and \( y \) are vectors of manifest variables. The components of \( \eta \) correspond to endogenous latent variables; the components of \( \xi \) correspond to exogenous latent variables. The endogenous and exogenous latent variables are connected by a system of linear equations (the structural model) with coefficient matrices \( B \) and \( \Gamma \) and an error vector \( \zeta \). It is assumed that matrix \( I - B \) is nonsingular. The random vectors \( y \) and \( x \) correspond to manifest variables that are related to the latent variables \( \eta \) and \( \xi \) by two systems of linear equations (the measurement model) with coefficients \( \Lambda_y \) and \( \Lambda_x \) and with measurement errors \( \varepsilon \) and \( \delta \).

Covariance Structure:

\[
C = J(I - A)^{-1}P(I - A)^{-1}'J' \\
A = \begin{pmatrix} 0 & 0 & \Lambda_y & 0 \\ 0 & 0 & 0 & \Lambda_x \\ 0 & 0 & B & \Gamma \\ 0 & 0 & 0 & 0 \end{pmatrix} \quad \text{and} \quad P = \begin{pmatrix} \Theta \varepsilon & \Theta \delta & \Psi \\ \Phi \end{pmatrix}
\]

with selection matrix \( J \), \( \Phi = \varepsilon \{ \xi \xi' \} \), \( \Psi = \varepsilon \{ \zeta \zeta' \} \), \( \Theta \delta = \varepsilon \{ \delta \delta' \} \), and \( \Theta \varepsilon = \varepsilon \{ \varepsilon \varepsilon' \} \).

Hierarchical Factor Analysis Models

First-order model:

\[
C = F_1P_1F_1' + U_1^2
\]

Second-order model:

\[
C = F_1F_2P_2F_2'F_1' + F_1U_1^2F_1' + U_1^2
\]

First-Order Autoregressive Longitudinal Factor Model

Example of McDonald (1980): \( k=3 \): Occasions of Measurement; \( n=3 \): Variables (Tests); \( m=2 \): Common Factors

\[
C = F_1F_2F_3LF_3^{-1}F_2^{-1}P(F_2^{-1})'(F_3^{-1})'L'F_3'F_2'F_1' + U^2
\]

\[
F_1 = \begin{pmatrix} B_1 \\ B_2 \\ B_3 \end{pmatrix}, \quad F_2 = \begin{pmatrix} I_2 \\ D_2 \end{pmatrix}, \quad F_3 = \begin{pmatrix} I_2 \\ I_2 \\ D_3 \end{pmatrix}
\]

\[
L = \begin{pmatrix} I_2 & o & o \\ I_2 & I_2 & o \\ I_2 & I_2 & I_2 \end{pmatrix}, \quad P = \begin{pmatrix} I_2 \\ S_2 \\ S_3 \end{pmatrix}, \quad U = \begin{pmatrix} U_{11} & U_{12} & U_{13} \\ U_{21} & U_{22} & U_{23} \\ U_{31} & U_{32} & U_{33} \end{pmatrix}
\]

\[
S_2 = I_2 - D_2^2, \quad S_3 = I_2 - D_3^2
\]

For more information on this model, see Example 17.6 on page 583.
A Structural Equation Example

This example from Wheaton et al. (1977) illustrates the relationships among the RAM, LINEQS, and LISREL models. Different structural models for these data are in Jöreskog and Sörbom (1985) and in Bentler (1985, p. 28). The data set contains covariances among six (manifest) variables collected from 932 people in rural regions of Illinois:

Variable 1: \( V_1, y_1 \) : Anomia 1967
Variable 2: \( V_2, y_2 \) : Powerlessness 1967
Variable 3: \( V_3, y_3 \) : Anomia 1971
Variable 4: \( V_4, y_4 \) : Powerlessness 1971
Variable 5: \( V_5, x_1 \) : Education (years of schooling)
Variable 6: \( V_6, x_2 \) : Duncan’s Socioeconomic Index (SEI)

It is assumed that anomia and powerlessness are indicators of an alienation factor and that education and SEI are indicators for a socioeconomic status (SES) factor. Hence, the analysis contains three latent variables:

Variable 7: \( F_1, \eta_1 \) : Alienation 1967
Variable 8: \( F_2, \eta_2 \) : Alienation 1971
Variable 9: \( F_3, \xi_1 \) : Socioeconomic Status (SES)

The following path diagram shows the structural model used in Bentler (1985, p. 29) and slightly modified in Jöreskog and Sörbom (1985, p. 56). In this notation for the path diagram, regression coefficients between the variables are indicated as one-headed arrows. Variances and covariances among the variables are indicated as two-headed arrows. Indicating error variances and covariances as two-headed arrows with the same source and destination (McArdle 1988; McDonald 1985) is helpful in transforming the path diagram to RAM model list input for the CALIS procedure.
Figure 17.1. Path Diagram of Stability and Alienation Example

Variables in Figure 17.1 are as follows:

Variable 1: \( V_1, y_1 \) : Anomia 1967
Variable 2: \( V_2, y_2 \) : Powerlessness 1967
Variable 3: \( V_3, y_3 \) : Anomia 1971
Variable 4: \( V_4, y_4 \) : Powerlessness 1971
Variable 5: \( V_5, x_1 \) : Education (years of schooling)
Variable 6: \( V_6, x_2 \) : Duncan’s Socioeconomic Index (SEI)
Variable 7: \( F_1, \eta_1 \) : Alienation 1967
Variable 8: \( F_2, \eta_2 \) : Alienation 1971
Variable 9: \( F_3, \xi_1 \) : Socioeconomic Status (SES)
**RAM Model**

The vector $v$ contains the six manifest variables $v_1 = V_1, \ldots, v_6 = V_6$ and the three latent variables $v_7 = F_1, v_8 = F_2, v_9 = F_3$. The vector $u$ contains the corresponding error variables $u_1 = E_1, \ldots, u_6 = E_6$ and $u_7 = D_1, u_8 = D_2, u_9 = D_3$. The path diagram corresponds to the following set of structural equations of the RAM model:

\[
\begin{align*}
    v_1 &= 1.000v_7 + u_1 \\
    v_2 &= 0.833v_7 + u_2 \\
    v_3 &= 1.000v_8 + u_3 \\
    v_4 &= 0.833v_8 + u_4 \\
    v_5 &= 1.000v_9 + u_5 \\
    v_6 &= \lambda v_9 + u_6 \\
    v_7 &= \gamma_1 v_9 + u_7 \\
    v_8 &= \beta v_7 + \gamma_2 v_9 + u_8 \\
    v_9 &= u_9
\end{align*}
\]

This gives the matrices $A$ and $P$ in the RAM model:

\[
A = \begin{pmatrix}
    o & o & o & o & o & 1.000 & o & o \\
    o & o & o & o & o & 0.833 & o & o \\
    o & o & o & o & o & o & 1.000 & o \\
    o & o & o & o & o & o & 0.833 & o \\
    o & o & o & o & o & o & o & \lambda \\
    o & o & o & o & o & o & o & \gamma_1 \\
    o & o & o & o & o & o & \beta & o & \gamma_2 \\
    o & o & o & o & o & o & o & o
\end{pmatrix}
\]

\[
P = \begin{pmatrix}
    \theta_1 & o & \theta_5 & o & o & o & o & o & o \\
    o & \theta_2 & o & \theta_5 & o & o & o & o & o \\
    \theta_5 & o & \theta_1 & o & o & o & o & o & o \\
    o & \theta_5 & o & \theta_2 & o & o & o & o & o \\
    o & o & o & o & \theta_3 & o & o & o & o \\
    o & o & o & o & o & \theta_4 & o & o & o \\
    o & o & o & o & o & o & \psi_1 & o & o \\
    o & o & o & o & o & o & o & \psi_2 & o \\
    o & o & o & o & o & o & o & o & \phi
\end{pmatrix}
\]

The RAM model input specification of this example for the CALIS procedure is in the “RAM Model Specification” section on page 410.

**LINEQS Model**

The vector $\eta$ contains the six endogenous manifest variables $V_1, \ldots, V_6$ and the two endogenous latent variables $F_1$ and $F_2$. The vector $\xi$ contains the exogenous
error variables \(E_1, \ldots, E_6, D_1,\) and \(D_2\) and the exogenous latent variable \(F_3\). The path diagram corresponds to the following set of structural equations of the LINEQS model:

\[
\begin{align*}
V_1 &= 1.0F_1 + E_1 \\
V_2 &= .833F_1 + E_2 \\
V_3 &= 1.0F_2 + E_3 \\
V_4 &= .833F_2 + E_4 \\
V_5 &= 1.0F_3 + E_5 \\
V_6 &= \lambda F_3 + E_6 \\
F_1 &= \gamma_1 F_3 + D_1 \\
F_2 &= \beta F_1 + \gamma_2 F_3 + D_2
\end{align*}
\]

This gives the matrices \(\beta, \gamma\) and \(\Phi\) in the LINEQS model:

\[
\beta = \begin{pmatrix}
1 & .833 & 1. \\
0 & 0 & 0 \\
0 & 0 & 0 \\
0 & 0 & 0 \\
0 & 0 & 0 \\
0 & 0 & 0
\end{pmatrix}, \quad \gamma = \begin{pmatrix}
1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 1 & 0 & 0 & \lambda & 1 \\
0 & 0 & 0 & 0 & 1 & 0 & \gamma_1 & 0 \\
0 & 0 & 0 & 0 & 0 & 1 & \gamma_2 & 0
\end{pmatrix}
\]

\[
\Phi = \begin{pmatrix}
\theta_1 & \theta_5 & \theta_5 & \theta_1 & \theta_5 & \theta_2 & \psi_1 & \phi \\
pos & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
pos & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
os & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
os & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
os & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
os & 0 & 0 & 0 & 0 & 0 & 0 & 0
\end{pmatrix}
\]

The LINEQS model input specification of this example for the CALIS procedure is in the section “LINEQS Model Specification” on page 409.

**LISREL Model**

The vector \(y\) contains the four endogenous manifest variables \(y_1 = V_1, \ldots, y_4 = V_4\), and the vector \(x\) contains the exogenous manifest variables \(x_1 = V_5\) and \(x_2 = V_6\). The vector \(e\) contains the error variables \(e_1 = E_1, \ldots, e_4 = E_4\) corresponding to \(y\), and the vector \(d\) contains the error variables \(d_1 = E_5\) and \(d_2 = E_6\) corresponding to \(x\). The vector \(\eta\) contains the endogenous latent variables (factors) \(\eta_1 = F_1\) and \(\eta_2 = F_2\), while the vector \(\xi\) contains the exogenous latent variable.
(factor) \( \xi_1 = F^3 \). The vector \( \zeta \) contains the errors \( \zeta_1 = D_1 \) and \( \zeta_2 = D_2 \) in the equations (disturbance terms) corresponding to \( \eta \). The path diagram corresponds to the following set of structural equations of the LISREL model:

\[
\begin{align*}
    y_1 &= 1.0\eta_1 + \epsilon_1 \\
    y_2 &= 0.833\eta_1 + \epsilon_2 \\
    y_3 &= 1.0\eta_2 + \epsilon_3 \\
    y_4 &= 0.833\eta_2 + \epsilon_4 \\
    x_1 &= 1.0\xi_1 + \delta_1 \\
    x_2 &= \lambda\xi_1 + \delta_2 \\
    \eta_1 &= \gamma_1\xi_1 + \zeta_1 \\
    \eta_2 &= \beta\eta_1 + \gamma_2\xi_1 + \zeta_2
\end{align*}
\]

This gives the matrices \( \Lambda_y, \Lambda_x, B, \Gamma, \) and \( \Phi \) in the LISREL model:

\[
\Lambda_y = \begin{pmatrix} 1 & 0 \\ 0.833 & 0 \\ 0 & 1.0 \\ 0 & 0.833 \end{pmatrix}, \quad \Lambda_x = \begin{pmatrix} 1.0 & 0 \\ \lambda & 1.0 \end{pmatrix}, \quad B = \begin{pmatrix} 0 & 0 \\ \beta & 0 \end{pmatrix}, \quad \Gamma = \begin{pmatrix} \gamma_1 \\ \gamma_2 \end{pmatrix}
\]

The CALIS procedure does not provide a LISREL model input specification. However, any model that can be specified by the LISREL model can also be specified by using the COSAN, LINEQS, or RAM model specifications in PROC CALIS.

**Getting Started**

There are four sets of statements available in the CALIS procedure to specify a model. Since a LISREL analysis can be performed easily by using a RAM, COSAN, or LINEQS statement, there is no specific LISREL input form available in the CALIS procedure.

For COSAN-style input, you can specify the following statements:

**COSAN** analysis model in matrix notation;

**MATRIX** definition of matrix elements;

**VARNAMES** names of additional variables;

**BOUNDS** boundary constraints;

**PARAMETERS** parameter names from program statements;
For linear equations input, you can specify the following statements:

```
LINEQS analysis model in equations notation;
      STD variance pattern;
      COV covariance pattern;
      BOUNDS boundary constraints;
      PARAMETERS parameter names from program statements;
```

For RAM-style input, you can specify the following statements:

```
RAM analysis model in list notation;
      VARNAMES names of latent and error variables;
      BOUNDS boundary constraints;
      PARAMETERS parameter names from program statements;
```

For (confirmatory) factor analysis input, you can specify the following statements:

```
FACTOR options;
      MATRIX definition of matrix elements;
      VARNAMES names of latent and residual variables;
      BOUNDS boundary constraints;
      PARAMETERS parameter names from program statements;
```

The model can also be obtained from an INRAM= data set, which is usually a version of an OUTRAM= data set produced by a previous PROC CALIS analysis (and possibly modified).

If no INRAM= data set is specified, you must use one of the four statements that defines the input form of the analysis model: COSAN, RAM, LINEQS, or FACTOR.

**COSAN Model Specification**

You specify the model for a generalized COSAN analysis with a COSAN statement and one or more MATRIX statements. The COSAN statement determines the name, dimension, and type (identity, diagonal, symmetric, upper, lower, general, inverse, and so forth) of each matrix in the model. You can specify the values of the constant elements in each matrix and give names and initial values to the elements that are to be estimated as parameters or functions of parameters using MATRIX statements. The resulting displayed output is in matrix form.

The following statements define the structural model of the alienation example as a COSAN model:

```
Cosan J(9, Ide) * A(9, Gen, Imi) * P(9, Sym);
Matrix A
    [ ,7] = 1. .833 5 * 0. Beta (.5) ,
    [ ,8] = 2 * 0. 1. .833 ,
    [ ,9] = 4 * 0. 1. Lamb Gam1-Gam2 (.5 2 * -.5);
```
Matrix P

\[
\begin{align*}
[1,1] &= \text{The1-The2 The1-The4} (6 \times 3.), \\
[7,7] &= \text{Psi1-Psi2 Phi} (2 \times 4. 6.), \\
[3,1] &= \text{The5} (.2), \\
[4,2] &= \text{The5} (.2);
\end{align*}
\]

The matrix model specified in the COSAN statement is the RAM model

\[
C = J(I - A)^{-1}P((I - A)^{-1}J')
\]

with selection matrix \( J \) and

\[
C = \mathcal{E}\{Jv'v'\}, \quad P = \mathcal{E}\{uu'\}
\]

The COSAN statement must contain only the matrices up to the central matrix \( P \) because of the symmetry of each matrix term in a COSAN model. Each matrix name is followed by one to three arguments in parentheses. The first argument is the number of columns. The second and third arguments are optional, and they specify the form of the matrix. The selection matrix \( J \) in the RAM model is specified by the 6 \( \times \) 9 identity (IDE) (sub)matrix \( J \) because the first six variables in vector \( v \) correspond to the six manifest variables in the data set. The 9 \( \times \) 9 parameter matrix \( A \) has a general (GEN) form and is used as \((I - A)^{-1}\) in the analysis, as indicated by the identity-minus-inverse (IMI) argument. The central 9 \( \times \) 9 matrix \( P \) is specified as a symmetric (SYM) matrix.

The MATRIX statement for matrix \( A \) specifies the values in columns 7, 8, and 9, which correspond to the three latent variables \( F_1, F_2, \) and \( F_3 \), in accordance with the RAM model. The other columns of \( A \) are assumed to be zero. The initial values for the parameter elements in \( A \) are chosen as in the path diagram to be

\[
\lambda = \beta = .5, \quad \gamma_1 = \gamma_2 = -.5
\]

In accordance with matrix \( P \) of the RAM model and the path model, the nine diagonal elements of matrix \( P \) are parameters with initial values

\[
\theta_1 = \theta_2 = \theta_3 = \theta_4 = 3, \quad \psi_1 = \psi_2 = 4, \quad \phi = 6
\]

There are also two off-diagonal elements in each triangle of \( P \) that are constrained to be equal, and they have an initial value of 0.2.

See the section “COSAN Model Statement” on page 437 for more information about the COSAN statement.

**LINEQS Model Specification**

You can also describe the model by a set of linear equations combined with variance and covariance specifications, using notation similar to that originally developed by P. Bentler for his EQS program. The displayed output can be in either equation form or matrix form.

The following statements define the structural model of the alienation example as a LINEQS model:
Chapter 17. The CALIS Procedure

Lineqs
V1 = F1 + E1,
V2 = .833 F1 + E2,
V3 = F2 + E3,
V4 = .833 F2 + E4,
V5 = F3 + E5,
V6 = Lamb (.5) F3 + E6,
F1 = Gam1(-.5) F3 + D1,
F2 = Beta (.5) F1 + Gam2(-.5) F3 + D2;

Std
E1-E6 = The1-The2 The1-The4 (6 * 3.),
D1-D2 = Psi1-Psi2 (2 * 4.),
F3 = Phi (6.) ;

Cov
E1 E3 = The5 (.2),
E4 E2 = The5 (.2);

The LINEQS statement shows the equations in the section “LINEQS Model” on page 405, except that in this case the coefficients to be estimated can be followed (optionally) by the initial value to use in the optimization process. If you do not specify initial values for the parameters in a LINEQS statement, PROC CALIS tries to assign these values automatically. The endogenous variables used on the left side can be manifest variables (with names that must be defined by the input data set) or latent variables (which must have names starting with F). The variables used on the right side can be manifest variables, latent variables (with names that must start with an F), or error variables (which must have names starting with an E or D). Commas separate the equations. The coefficients to be estimated are indicated by names. If no name is used, the coefficient is constant, either equal to a specified number or, if no number is used, equal to 1. The VAR statement in Bentler’s notation is replaced here by the STD statement, because the VAR statement in PROC CALIS defines the subset of manifest variables in the data set to be analyzed. The variable names used in the STD or COV statement must be exogenous (that is, they should not occur on the left side of any equation). The STD and COV statements define the diagonal and off-diagonal elements in the \( \Phi \) matrix. The parameter specifications in the STD and COV statements are separated by commas. Using \( k \) variable names on the left of an equal sign in a COV statement means that the parameter list on the right side refers to all \( k(k-1)/2 \) distinct variable pairs in the \( \Phi \) matrix. Identical coefficient names indicate parameters constrained to be equal. You can also use prefix names to specify those parameters for which you do not need a precise name in any parameter constraint.

See the section “LINEQS Model Statement” on page 447 for more information about the precise syntax rules for a LINEQS statement.

**RAM Model Specification**

The RAM model allows a path diagram to be transcribed into a RAM statement in list form. The displayed output from the RAM statement is in matrix or list form.

The following statement defines the structural model of the alienation example as a RAM model:
You must assign numbers to the nodes in the path diagram. In the path diagram of Figure 17.1, the boxes corresponding to the six manifest variables \(V_1, \ldots, V_6\) are assigned the number of the variable in the covariance matrix (1, \ldots, 6); the circles corresponding to the three latent variables \(F_1, F_2, \) and \(F_3\) are given the numbers 7, 8, and 9. The path diagram contains 20 paths between the nine nodes; nine of the paths are one-headed arrows and eleven are two-headed arrows.

The RAM statement contains a list of items separated by commas. Each item corresponds to an arrow in the path diagram. The first entry in each item is the number of arrow heads (matrix number), the second entry shows where the arrow points to (row number), the third entry shows where the arrow comes from (column number), the fourth entry gives the (initial) value of the coefficient, and the fifth entry assigns a name if the path represents a parameter rather than a constant. If you specify the fifth entry as a parameter name, then the fourth list entry can be omitted, since PROC CALIS tries to assign an initial value to this parameter.

See the section “RAM Model Statement” on page 443 for more information about the RAM statement.

**FACTOR Model Specification**

You can specify the FACTOR statement to compute factor loadings \(F\) and unique variances \(U\) of an exploratory or confirmatory first-order factor (or component) analysis. By default, the factor correlation matrix \(P\) is an identity matrix.

\[
C = FF' + U, \quad U = diag
\]

For a first-order confirmatory factor analysis, you can use MATRIX statements to define elements in the matrices \(F, P, \) and \(U\) of the more general model

\[
C = FF' + U, \quad P = P', \quad U = diag
\]
To perform a component analysis, specify the COMPONENT option to constrain the matrix $U$ to a zero matrix; that is, the model is replaced by

$$C = FF'$$

Note that the rank of $FF'$ is equal to the number $m$ of components in $F$, and if $m$ is smaller than the number of variables in the moment matrix $C$, the matrix of predicted model values is singular and maximum likelihood estimates for $F$ cannot be computed. You should compute ULS estimates in this case.

The HEYWOOD option constrains the diagonal elements of $U$ to be nonnegative; that is, the model is replaced by

$$C = FF' + U^2, \quad U = diag$$

If the factor loadings are unconstrained, they can be orthogonally rotated by one of the following methods:

- principal axes rotation
- quartimax
- varimax
- equamax
- parsimax

The most common approach to factor analysis consists of two steps:

1. Obtain estimates for factor loadings and unique variances.
2. Apply an orthogonal or oblique rotation method.

Most programs of factor analysis do not provide standard errors for the rotated factor loadings. PROC CALIS enables you to specify general linear and nonlinear equality and inequality constraints using the LINCON and NLINCON statements. You can specify the NLINCON statement to estimate orthogonal or oblique rotated factor loadings; refer to Browne and Du Toit (1992).

For default (exploratory) factor analysis, PROC CALIS computes initial estimates. If you use a MATRIX statement together with a FACTOR model specification, initial values are generally computed by McDonald’s (McDonald and Hartmann 1992) method or are set by the START= option. See the section “FACTOR Model Statement” on page 453 and Example 17.3 for more information about the FACTOR statement.

**Constrained Estimation**

- Simple equality constraints, $x_i = c_i$, $c_i = const$, and $x_i = x_j$, can be defined in each model by specifying constants or using the same name for parameters constrained to be equal.
• **BOUNDS statement**: You can specify boundary constraints, \( l_i \leq x_i \leq u_i \), \( l_i, u_i = \text{const} \), with the BOUNDS statement for the COSAN, LINEQS, and RAM models and in connection with an INRAM= data set. There may be serious convergence problems if negative values appear in the diagonal locations (variances) of the central model matrices during the minimization process. You can use the BOUNDS statement to constrain these parameters to have nonnegative values.

• **LINCON statement**: You can specify general linear equality and inequality constraints of the parameter estimates with the LINCON statement or by using an INEST= data set. The variables listed in the LINCON statements must be (a subset of) the model parameters. All optimization methods can be used for linear constrained optimization.

• **NLINCON statement**: You can specify general nonlinear equality and inequality constraints of the parameter estimates with the NLINCON statement. The syntax of the NLINCON statement is almost the same as that for the BOUNDS statement with the exception that the BOUNDS statement can contain only names of the model parameters. However, the variables listed in the NLINCON statement can be defined by program statements. Only the quasi-Newton optimization method can be used for nonlinearly constrained optimization.

• **Reparameterizing the Model**: Complex linear equality and inequality constraints can be defined by means of program statements similar to those used in the DATA step. In this case, some of the parameters \( x_i \) are not elements of the matrices \( G_{kj} \) and \( Q_k \) but are instead defined in a PARAMETERS statement. Elements of the model matrices can then be computed by program statements as functions of parameters in the PARAMETERS statement. This approach is similar to the classical COSAN program of R. McDonald, implemented by C. Fraser (McDonald 1978, 1980). One advantage of the CALIS procedure is that you need not supply code for the derivatives of the specified functions. The analytic derivatives of the user-written functions are computed automatically by PROC CALIS. The specified functions must be continuous and have continuous first-order partial derivatives. See the “SAS Program Statements” section on page 474 and the “Constrained Estimation Using Program Code” section on page 520 for more information about imposing linear and nonlinear restrictions on parameters by using program statements.

Although much effort has been made to implement reliable and numerically stable optimization methods, no practical algorithm exists that can always find the global optimum of a nonlinear function, especially when there are nonlinear constraints.
Syntax

**PROC CALIS** < *options*> ;
**COSAN** matrix model ;
**MATRIX** matrix elements ;
**VARNAMES** variables ;
**LINEQS** model equations ;
**STD** variance pattern ;
**COV** covariance pattern ;
**RAM** model list ;
**VARNAMES** variables ;
**FACTOR** < *options*> ;
**MATRIX** matrix elements ;
**VARNAMES** variables ;
**BOUNDS** boundary constraints ;
**BY** variables ;
**FREQ** variable ;
**LINCON** linear constraints ;
**NLLINCON** nonlinear constraints ;
**NLOPTOPTIONS** optimization options ;
**PARAMETERS** parameters ;
**PARTIAL** variables ;
**STRUCTEQ** variables ;
**VAR** variables ;
**WEIGHT** variable ;
*program statements*

- If no INRAM= data set is specified, one of the four statements that defines the input form of the analysis model, COSAN, LINEQS, RAM, or FACTOR, must be used.
- The MATRIX statement can be used multiple times for the same or different matrices along with a COSAN or FACTOR statement. If the MATRIX statement is used multiple times for the same matrix, later definitions override earlier ones.
- The STD and COV statements can be used only with the LINEQS model statement.
- You can formulate a generalized COSAN model using a COSAN statement. MATRIX statements can be used to define the elements of a matrix used in the COSAN statement. The input notation resembles the COSAN program of R. McDonald and C. Fraser (McDonald 1978, 1980).
- The RAM statement uses a simple list input that is especially suitable for describing J. McArdle’s RAM analysis model (McArdle 1980, McArdle and McDonald 1984) for causal and path analysis problems.
- The LINEQS statement formulates the analysis model by means of a system of linear equations similar to P. Bentler’s (1989) EQS program notation. The
STD and COV statements can be used to define the variances and covariances corresponding to elements of matrix $\Phi$ in the LINEQS model.

- A FACTOR statement can be used to compute a first-order exploratory or confirmatory factor (or component) analysis. The analysis of a simple exploratory factor analysis model performed by PROC CALIS is not as efficient as one performed by the FACTOR procedure. The CALIS procedure is designed for more general structural problems, and it needs significantly more computation time for a simple unrestricted factor or component analysis than does PROC FACTOR.

- You can add program statements to impose linear or nonlinear constraints on the parameters if you specify the model by means of a COSAN, LINEQS, or RAM statement. The PARAMETERS statement defines additional parameters that are needed as independent variables in your program code and that belong to the set of parameters to be estimated. Variable names used in the program code should differ from the preceding statement names. The code should respect the syntax rules of SAS statements usually used in the DATA step. See the “SAS Program Statements” section on page 474 for more information.

- The BOUNDS statement can be used to specify simple lower and upper boundary constraints for the parameters.

- You can specify general linear equality and inequality constraints with the LINCON statement (or via an INEST= data set). The NLINCON statement can be used to specify general nonlinear equality and inequality constraints by referring to nonlinear functions defined by program statements.

- The VAR, PARTIAL, WEIGHT, FREQ, and BY statements can be used in the same way as in other procedures, for example, the FACTOR or PRINCOMP procedure. You can select a subset of the input variables to analyze with the VAR statement. The PARTIAL statement defines a set of input variables that are chosen as partial variables for the analysis of a matrix of partial correlations or covariances. The BY statement specifies groups in which separate covariance structure analyses are performed.

### PROC CALIS Statement

```
PROC CALIS < options >;
```

This statement invokes the procedure. The options available with the PROC CALIS statement are summarized in Table 17.1 and discussed in the following six sections.

#### Table 17.1. PROC CALIS Statement Options

<table>
<thead>
<tr>
<th>Data Set Options</th>
<th>Short Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DATA=</td>
<td>input data set</td>
</tr>
<tr>
<td>INEST=</td>
<td>input initial values, constraints</td>
</tr>
<tr>
<td>INRAM=</td>
<td>input model</td>
</tr>
<tr>
<td>INWGT=</td>
<td>input weight matrix</td>
</tr>
</tbody>
</table>
### Data Set Options

<table>
<thead>
<tr>
<th>Option</th>
<th>Short Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>OUTTEST=</td>
<td>covariance matrix of estimates</td>
</tr>
<tr>
<td>OUTJAC</td>
<td>Jacobian into OUTTEST= data set</td>
</tr>
<tr>
<td>OUTRAM=</td>
<td>output model</td>
</tr>
<tr>
<td>OUTSTAT=</td>
<td>output statistic</td>
</tr>
<tr>
<td>OUTWGT=</td>
<td>output weight matrix</td>
</tr>
</tbody>
</table>

### Data Processing

<table>
<thead>
<tr>
<th>Option</th>
<th>Short Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AUGMENT</td>
<td>analyzes augmented moment matrix</td>
</tr>
<tr>
<td>COVARIANCE</td>
<td>analyzes covariance matrix</td>
</tr>
<tr>
<td>EDF=</td>
<td>defines nobs by number error df</td>
</tr>
<tr>
<td>NOBS=</td>
<td>defines number of observations nobs</td>
</tr>
<tr>
<td>NOINT</td>
<td>analyzes uncorrected moments</td>
</tr>
<tr>
<td>RDF=</td>
<td>defines nobs by number regression df</td>
</tr>
<tr>
<td>RIDGE</td>
<td>specifies ridge factor for moment matrix</td>
</tr>
<tr>
<td>UCORR</td>
<td>analyzes uncorrected CORR matrix</td>
</tr>
<tr>
<td>UCOV</td>
<td>analyzes uncorrected COV matrix</td>
</tr>
<tr>
<td>VARDEF=</td>
<td>specifies variance divisor</td>
</tr>
</tbody>
</table>

### Estimation Methods

<table>
<thead>
<tr>
<th>Option</th>
<th>Short Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>METHOD=</td>
<td>estimation method</td>
</tr>
<tr>
<td>ASYCOV=</td>
<td>formula of asymptotic covariances</td>
</tr>
<tr>
<td>DFREDUCE=</td>
<td>reduces degrees of freedom</td>
</tr>
<tr>
<td>G4=</td>
<td>algorithm for STDERR</td>
</tr>
<tr>
<td>NODIAG</td>
<td>excludes diagonal elements from fit</td>
</tr>
<tr>
<td>WPENALTY=</td>
<td>penalty weight to fit correlations</td>
</tr>
<tr>
<td>WRIDGE=</td>
<td>ridge factor for weight matrix</td>
</tr>
</tbody>
</table>

### Optimization Techniques

<table>
<thead>
<tr>
<th>Option</th>
<th>Short Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>TECHNIQUE=</td>
<td>minimization method</td>
</tr>
<tr>
<td>UPDATE=</td>
<td>update technique</td>
</tr>
<tr>
<td>LINESEARCH=</td>
<td>line-search method</td>
</tr>
<tr>
<td>FCONV=</td>
<td>function convergence criterion</td>
</tr>
<tr>
<td>GCONV=</td>
<td>gradient convergence criterion</td>
</tr>
<tr>
<td>INSTEP=</td>
<td>initial step length (RADIUS=, SALPHA=)</td>
</tr>
<tr>
<td>LSPRECISION=</td>
<td>line-search precision (SPRECISION=)</td>
</tr>
<tr>
<td>MAXFUNC=</td>
<td>max number function calls</td>
</tr>
<tr>
<td>MAXITER=</td>
<td>max number iterations</td>
</tr>
</tbody>
</table>

### Displayed Output Options

<table>
<thead>
<tr>
<th>Option</th>
<th>Short Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>KURTOSIS</td>
<td>compute and display kurtosis</td>
</tr>
<tr>
<td>MODIFICATION</td>
<td>modification indices</td>
</tr>
<tr>
<td>NOMOD</td>
<td>no modification indices</td>
</tr>
<tr>
<td>NOPRINT</td>
<td>suppresses the displayed output</td>
</tr>
<tr>
<td>PALL</td>
<td>all displayed output (ALL)</td>
</tr>
<tr>
<td>PCORR</td>
<td>analyzed and estimated moment matrix</td>
</tr>
</tbody>
</table>
## Displayed Output Options

<table>
<thead>
<tr>
<th>Option</th>
<th>Short Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>PCOVES</td>
<td>covariance matrix of estimates</td>
</tr>
<tr>
<td>PDETERM</td>
<td>determination coefficients</td>
</tr>
<tr>
<td>PESTIM</td>
<td>parameter estimates</td>
</tr>
<tr>
<td>PINITIAL</td>
<td>pattern and initial values</td>
</tr>
<tr>
<td>PJACPAT</td>
<td>displays structure of variable and constant elements of the Jacobian matrix</td>
</tr>
<tr>
<td>PLATCOV</td>
<td>latent variable covariances, scores</td>
</tr>
<tr>
<td>PREDET</td>
<td>displays predetermined moment matrix</td>
</tr>
<tr>
<td>PRIMAT</td>
<td>displays output in matrix form</td>
</tr>
<tr>
<td>PRINT</td>
<td>adds default displayed output</td>
</tr>
<tr>
<td>PRIVVEC</td>
<td>displays output in vector form</td>
</tr>
<tr>
<td>PSHORT</td>
<td>reduces default output (SHORT)</td>
</tr>
<tr>
<td>PSUMMARY</td>
<td>displays only fit summary (SUMMARY)</td>
</tr>
<tr>
<td>PWEIGHT</td>
<td>weight matrix</td>
</tr>
<tr>
<td>RESIDUAL=</td>
<td>residual matrix and distribution</td>
</tr>
<tr>
<td>SIMPLE</td>
<td>univariate statistics</td>
</tr>
<tr>
<td>STDERR</td>
<td>standard errors</td>
</tr>
<tr>
<td>TOTTEFF</td>
<td>displays total and indirect effects</td>
</tr>
</tbody>
</table>

## Miscellaneous Options

<table>
<thead>
<tr>
<th>Option</th>
<th>Short Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ALPHAECV=</td>
<td>probability Browne &amp; Cudeck ECV</td>
</tr>
<tr>
<td>ALPHARMS=</td>
<td>probability Steiger &amp; Lind RMSEA</td>
</tr>
<tr>
<td>BIASKUR</td>
<td>biased skewness and kurtosis</td>
</tr>
<tr>
<td>DEMPHAS=</td>
<td>emphasizes diagonal entries</td>
</tr>
<tr>
<td>FDCODE</td>
<td>uses numeric derivatives for code</td>
</tr>
<tr>
<td>HESSALG=</td>
<td>algorithm for Hessian</td>
</tr>
<tr>
<td>NOSTDERR</td>
<td>computes no standard errors</td>
</tr>
<tr>
<td>RANDOM=</td>
<td>randomly generated initial values</td>
</tr>
<tr>
<td>SINGULAR=</td>
<td>singularity criterion</td>
</tr>
<tr>
<td>ASINGULAR=</td>
<td>absolute singularity information matrix</td>
</tr>
<tr>
<td>COVSING=</td>
<td>singularity tolerance of information matrix</td>
</tr>
<tr>
<td>MSINGULAR=</td>
<td>relative M singularity of information matrix</td>
</tr>
<tr>
<td>VSINGULAR=</td>
<td>relative V singularity of information matrix</td>
</tr>
<tr>
<td>SLMW=</td>
<td>probability limit for Wald test</td>
</tr>
<tr>
<td>START=</td>
<td>constant initial values</td>
</tr>
</tbody>
</table>

### Data Set Options

**DATA=SAS-data-set**

specifies an input data set that can be an ordinary SAS data set or a specially structured TYPE=CORR, TYPE=COV, TYPE=UCORR, TYPE=UCOV, TYPE=SSCP, or TYPE=FACTOR SAS data set, as described in the section “Input Data Sets” on page 476. If the DATA= option is omitted, the most recently created SAS data set is used.
INEST | INVAR | ESTDATA= SAS-data-set
specifies an input data set that contains initial estimates for the parameters used in the
optimization process and can also contain boundary and general linear constraints on
the parameters. If the model did not change too much, you can specify an OUTEST= data set from a previous PROC CALIS analysis. The initial estimates are taken from
the values of the PARMS observation.

INRAM= SAS-data-set
specifies an input data set that contains in RAM list form all information needed to
specify an analysis model. The INRAM= data set is described in the section “In-
put Data Sets” on page 476. Typically, this input data set is an OUTRAM= data set
(possibly modified) from a previous PROC CALIS analysis. If you use an INRAM=
data set to specify the analysis model, you cannot use the model specification state-
ments COSAN, MATRIX, RAM, LINEQS, STD, COV, FACTOR, or VARNAMES,
but you can use the BOUNDS and PARAMETERS statements and program state-
ments. If the INRAM= option is omitted, you must define the analysis model with a
COSAN, RAM, LINEQS, or FACTOR statement.

INWGT= SAS-data-set
specifies an input data set that contains the weight matrix \( W \) used in generalized
least-squares (GLS), weighted least-squares (WLS, ADF), or diagonally weighted
least-squares (DWLS) estimation. If the weight matrix \( W \) defined by an INWGT=
data set is not positive definite, it can be ridged using the WRIDGE= option. See the
section “Estimation Criteria” on page 490 for more information. If no INWGT= data
set is specified, default settings for the weight matrices are used in the estimation
process. The INWGT= data set is described in the section “Input Data Sets” on
page 476. Typically, this input data set is an OUTWGT= data set from a previous
PROC CALIS analysis.

OUTEST | OUTVAR= SAS-data-set
creates an output data set containing the parameter estimates, their gradient, Hessian
matrix, and boundary and linear constraints. For METHOD=ML, METHOD=GLS,
and METHOD=WLS, the OUTEST= data set also contains the information matrix,
the approximate covariance matrix of the parameter estimates (generalized inverse
of information matrix), and approximate standard errors. If linear or nonlinear equal-
ity or active inequality constraints are present, the Lagrange multiplier estimates of
the active constraints, the projected Hessian, and the Hessian of the Lagrange func-
tion are written to the data set. The OUTEST= data set also contains the Jacobian if
the OUTJAC option is used.

The OUTEST= data set is described in the section “OUTEST= SAS-data-set” on
page 480. If you want to create a permanent SAS data set, you must specify a two-
level name. Refer to the chapter titled “SAS Data Files” in SAS Language Reference: Concepts for more information on permanent data sets.

OUTJAC
writes the Jacobian matrix, if it has been computed, to the OUTEST= data set. This
is useful when the information and Jacobian matrices need to be computed for other
analyses.
**OUTSTAT**=*SAS-data-set*

creates an output data set containing the BY group variables, the analyzed covariance or correlation matrices, and the predicted and residual covariance or correlation matrices of the analysis. You can specify the correlation or covariance matrix in an OUTSTAT= data set as an input DATA= data set in a subsequent analysis by PROC CALIS. The OUTSTAT= data set is described in the section “OUTSTAT= SAS-data-set” on page 487. If the model contains latent variables, this data set also contains the predicted covariances between latent and manifest variables and the latent variables scores regression coefficients (see the PLATCOV option on page 432). If the FACTOR statement is used, the OUTSTAT= data set also contains the rotated and unrotated factor loadings, the unique variances, the matrix of factor correlations, the transformation matrix of the rotation, and the matrix of standardized factor loadings.

You can specify the latent variable score regression coefficients with PROC SCORE to compute factor scores.

If you want to create a permanent SAS data set, you must specify a two-level name. Refer to the chapter titled “SAS Data Files” in *SAS Language Reference: Concepts* for more information on permanent data sets.

**OUTRAM**=*SAS-data-set*

creates an output data set containing the model information for the analysis, the parameter estimates, and their standard errors. An OUTRAM= data set can be used as an input INRAM= data set in a subsequent analysis by PROC CALIS. The OUTRAM= data set also contains a set of fit indices; it is described in more detail in the section “OUTRAM= SAS-data-set” on page 484. If you want to create a permanent SAS data set, you must specify a two-level name. Refer to the chapter titled “SAS Data Files” in *SAS Language Reference: Concepts* for more information on permanent data sets.

**OUTWGT**=*SAS-data-set*

creates an output data set containing the weight matrix $W$ used in the estimation process. You cannot create an OUTWGT= data set with an unweighted least-squares or maximum likelihood estimation. The fit function in GLS, WLS (ADF), and DWLS estimation contain the inverse of the (Cholesky factor of the) weight matrix $W$ written in the OUTWGT= data set. The OUTWGT= data set contains the weight matrix on which the WRIDGE= and the WPENALTY= options are applied. An OUTWGT= data set can be used as an input INWGT= data set in a subsequent analysis by PROC CALIS. The OUTWGT= data set is described in the section “OUTWGT= SAS-data-set” on page 489. If you want to create a permanent SAS data set, you must specify a two-level name. Refer to the chapter titled “SAS Data Files” in *SAS Language Reference: Concepts* for more information on permanent data sets.

**Data Processing Options**

**AUGMENT** | **AUG**

analyzes the augmented correlation or covariance matrix. Using the AUG option is equivalent to specifying UCORR (NOINT but not COV) or UCOV (NOINT and COV) for a data set that is augmented by an intercept variable INTERCEPT that has constant values equal to 1. The variable INTERCEP can be used instead of the default INTERCEPT only if you specify the SAS option OPTIONS VALIDVARNAMES=V6.
The dimension of an augmented matrix is one higher than that of the corresponding correlation or covariance matrix. The AUGMENT option is effective only if the data set does not contain a variable called INTERCEPT and if you specify the UCOV, UCORR, or NOINT option.

**CAUTION:** The INTERCEPT variable is included in the moment matrix as the variable with number \( n + 1 \). Using the RAM model statement assumes that the first \( n \) variable numbers correspond to the \( n \) manifest variables in the input data set. Therefore, specifying the AUGMENT option assumes that the numbers of the latent variables used in the RAM or path model have to start with number \( n + 2 \).

**COVARIANCE | COV**

analyzes the covariance matrix instead of the correlation matrix. By default, PROC CALIS (like the FACTOR procedure) analyzes a correlation matrix. If the DATA= input data set is a valid TYPE=CORR data set (containing a correlation matrix and standard deviations), using the COV option means that the covariance matrix is computed and analyzed.

**DFE | EDF=**

makes the effective number of observations \( n + i \), where \( i = 0 \) if the NOINT, UCORR, or UCOV option is specified without the AUGMENT option or where \( i = 1 \) otherwise. You can also use the NOBS= option to specify the number of observations.

**DFR | RDF=**

makes the effective number of observations the actual number of observations minus the RDF= value. The degree of freedom for the intercept should not be included in the RDF= option. If you use PROC CALIS to compute a regression model, you can specify RDF= number-of-repressor-variables to get approximate standard errors equal to those computed by PROC REG.

**NOBS=**

specifies the number of observations. If the DATA= input data set is a raw data set, \( nobs \) is defined by default to be the number of observations in the raw data set. The NOBS= and EDF= options override this default definition. You can use the RDF= option to modify the \( nobs \) specification. If the DATA= input data set contains a covariance, correlation, or scalar product matrix, you can specify the number of observations either by using the NOBS=, EDF=, and RDF= options in the PROC CALIS statement or by including a \_TYPE=’N’ \ observation in the DATA= input data set.

**NOINT**

specifies that no intercept be used in computing covariances and correlations; that is, covariances or correlations are not corrected for the mean. You can specify this option (or UCOV or UCORR) to analyze mean structures in an uncorrected moment matrix, that is, to compute intercepts in systems of structured linear equations (see Example 17.2). The term NOINT is misleading in the case that an uncorrected covariance or correlation matrix is analyzed containing a constant (intercept) variable that is used in the analysis model. The degrees of freedom used in the variance divisor (specified by the VARDEF= option) and some of the assessment of the fit function (see the section “Assessment of Fit” on page 495) depend on whether an intercept
variable is included in the model (the intercept is used in computing the corrected covariance or correlation matrix or is used as a variable in the uncorrected covariance or correlation matrix to estimate mean structures) or not included (an uncorrected covariance or correlation matrix is used that does not contain a constant variable).

**RIDGE**

\[ \text{RIDGE} = r \]

defines a ridge factor \( r \) for the diagonal of the moment matrix \( S \) that is analyzed. The matrix \( S \) is transformed to

\[ S \rightarrow \tilde{S} = S + r(diag(S)) \]

If you do not specify \( r \) in the RIDGE option, PROC CALIS tries to ridge the moment matrix \( S \) so that the smallest eigenvalue is about \( 10^{-3} \).

**CAUTION:** The moment matrix in the OUTSTAT= output data set does not contain the ridged diagonal.

**UCORR**

analyzes the uncorrected correlation matrix instead of the correlation matrix corrected for the mean. Using the UCORR option is equivalent to specifying the NOINT option but not the COV option.

**UCOV**

analyzes the uncorrected covariance matrix instead of the covariance matrix corrected for the mean. Using the UCOV option is equivalent to specifying both the COV and NOINT options. You can specify this option to analyze mean structures in an uncorrected covariance matrix, that is, to compute intercepts in systems of linear structural equations (see Example 17.2).

**VARDEF=** **DF | N | WDF | WEIGHT | WGT**

specifies the divisor used in the calculation of covariances and standard deviations. The default value is VARDEF=DF. The values and associated divisors are displayed in the following table, where \( i = 0 \) if the NOINT option is used and \( i = 1 \) otherwise and where \( k \) is the number of partial variables specified in the PARTIAL statement. Using an intercept variable in a mean structure analysis, by specifying the AUGMENT option, includes the intercept variable in the analysis. In this case, \( i = 1 \).

When a WEIGHT statement is used, \( w_j \) is the value of the WEIGHT variable in the \( j \)th observation, and the summation is performed only over observations with positive weight.

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
<th>Divisor</th>
</tr>
</thead>
<tbody>
<tr>
<td>DF</td>
<td>degrees of freedom</td>
<td>( N - k - i )</td>
</tr>
<tr>
<td>N</td>
<td>number of observations</td>
<td>( N )</td>
</tr>
<tr>
<td>WDF</td>
<td>sum of weights DF</td>
<td>( \sum_{i}^{N} w_j - k - i )</td>
</tr>
<tr>
<td>WEIGHT</td>
<td>sum of weights</td>
<td>( \sum_{j}^{N} w_j )</td>
</tr>
</tbody>
</table>

**Estimation Methods**

The default estimation method is maximum likelihood (METHOD=ML), assuming a multivariate normal distribution of the observed variables. The two-stage estimation methods METHOD=LSML, METHOD=LSGLS, METHOD=LSWLS, and METHOD=LSDWLS first compute unweighted least-squares estimates of the model
parameters and their residuals. Afterward, these estimates are used as initial
values for the optimization process to compute maximum likelihood, generalized
least-squares, weighted least-squares, or diagonally weighted least-squares parameter esti-
mates. You can do the same thing by using an OUTRAM= data set with least-squares
estimates as an INRAM= data set for a further analysis to obtain the second set of pa-
parameter estimates. This strategy is also discussed in the section “Use of Optimization
Techniques” on page 510. For more details, see the “Estimation Criteria” section on
page 490.

**METHOD | MET=name**
specifies the method of parameter estimation. The default is METHOD=ML. Valid
values for name are as follows:

- **ML | M | MAX**
  performs normal-theory maximum likelihood parameter
  estimation. The ML method requires a nonsingular covariance or correlation matrix.

- **GLS | G**
  performs generalized least-squares parameter estimation.
  If no INWGT= data set is specified, the GLS method
  uses the inverse sample covariance or correlation matrix
  as weight matrix $W$. Therefore, METHOD=GLS requires
  a nonsingular covariance or correlation matrix.

- **WLS | W | ADF**
  performs weighted least-squares parameter estimation. If
  no INWGT= data set is specified, the WLS method uses
  the inverse matrix of estimated asymptotic covariances of
  the sample covariance or correlation matrix as the weight
  matrix $W$. In this case, the WLS estimation method
  is equivalent to Browne’s (1982, 1984) asymptotically
distribution-free estimation. The WLS method requires a
  nonsingular weight matrix.

- **DWLS | D**
  performs diagonally weighted least-squares parameter es-
  timation. If no INWGT= data set is specified, the DWLS
  method uses the inverse diagonal matrix of asymptotic
  variances of the input sample covariance or correlation ma-
  trix as the weight matrix $W$. The DWLS method requires
  a nonsingular diagonal weight matrix.

- **ULS | LS | U**
  performs unweighted least-squares parameter estimation.

- **LSML | LSM | LSMAX**
  performs unweighted least-squares followed by normal-
  theory maximum likelihood parameter estimation.

- **LSGLS | LSG**
  performs unweighted least-squares followed by general-
  ized least-squares parameter estimation.

- **LSWLS | LSW | LSADF**
  performs unweighted least-squares followed by weighted
  least-squares parameter estimation.

- **LSDWLS | LSD**
  performs unweighted least-squares followed by diagonally
  weighted least-squares parameter estimation.
NONE | NO | N  
uses no estimation method. This option is suitable for checking the validity of the input information and for displaying the model matrices and initial values.

ASYCOV | ASC=name

specifies the formula for asymptotic covariances used in the weight matrix \( W \) for WLS and DWLS estimation. The ASYCOV option is effective only if METHOD=WLS or METHOD=DWLS and no INWGT= input data set is specified. The following formulas are implemented:

BIASED: Browne’s (1984) formula (3.4)
biased asymptotic covariance estimates; the resulting weight matrix is at least positive semidefinite. This is the default for analyzing a covariance matrix.

UNBIASED: Browne’s (1984) formula (3.8)
asymptotic covariance estimates corrected for bias; the resulting weight matrix can be indefinite (that is, can have negative eigenvalues), especially for small \( N \).

CORR: Browne and Shapiro’s (1986) formula (3.2)
(identical to DeLeeuw’s (1983) formulas (2,3,4)) the asymptotic variances of the diagonal elements are set to the reciprocal of the value \( r \) specified by the WPENALTY= option (default: \( r = 100 \)). This formula is the default for analyzing a correlation matrix.

CAUTION: Using the WLS and DWLS methods with the ASYCOV=CORR option means that you are fitting a correlation (rather than a covariance) structure. Since the fixed diagonal of a correlation matrix for some models does not contribute to the model’s degrees of freedom, you can specify the DFREDUCE= option to reduce the degrees of freedom by the number of manifest variables used in the model. See the section “Counting the Degrees of Freedom” on page 522 for more information.

DFREDUCE | DFRED=i

reduces the degrees of freedom of the \( \chi^2 \) test by \( i \). In general, the number of degrees of freedom is the number of elements of the lower triangle of the predicted model matrix \( C \), \( n(n+1)/2 \), minus the number of parameters, \( t \). If the NODIAG option is used, the number of degrees of freedom is additionally reduced by \( n \). Because negative values of \( i \) are allowed, you can also increase the number of degrees of freedom by using this option. If the DFREDUCE= or NODIAG option is used in a correlation structure analysis, PROC CALIS does not additionally reduce the degrees of freedom by the number of constant elements in the diagonal of the predicted model matrix, which is otherwise done automatically. See the section “Counting the Degrees of Freedom” on page 522 for more information.

G4=i

specifies the algorithm to compute the approximate covariance matrix of parameter estimates used for computing the approximate standard errors and modification indices when the information matrix is singular. If the number of parameters \( t \) used in the model you analyze is smaller than the value of \( i \), the time-expensive
Moore-Penrose (G4) inverse of the singular information matrix is computed by eigenvalue decomposition. Otherwise, an inexpensive pseudo (G1) inverse is computed by sweeping. By default, \( i = 60 \). For more details, see the section “Estimation Criteria” on page 490.

**NODIAG | NODI**

omits the diagonal elements of the analyzed correlation or covariance matrix from the fit function. This option is useful only for special models with constant error variables. The NODIAG option does not allow fitting those parameters that contribute to the diagonal of the estimated moment matrix. The degrees of freedom are automatically reduced by \( n \). A simple example for the usefulness of the NODIAG option is the fit of the first-order factor model, \( S = FF' + U^2 \). In this case, you do not have to estimate the diagonal matrix of unique variances \( U^2 \) that are fully determined by \( diag(S - FF') \).

**WPENALTY | WPEN=r**

specifies the penalty weight \( r \geq 0 \) for the WLS and DWLS fit of the diagonal elements of a correlation matrix (constant 1s). The criterion for weighted least-squares estimation of a correlation structure is

\[
F_{WLS} = \sum_{i=2}^{n} \sum_{j=1}^{n-1} \sum_{k=2}^{n} \sum_{l=1}^{k-1} w_{ij,kl}(s_{ij} - c_{ij})(s_{kl} - c_{kl}) + r \sum_{i=2}^{n} (s_{ii} - c_{ii})^2
\]

where \( r \) is the penalty weight specified by the WPENALTY=r option and the \( w_{ij,kl} \) are the elements of the inverse of the reduced \( (n(n-1)/2) \times (n(n-1)/2) \) weight matrix that contains only the nonzero rows and columns of the full weight matrix \( W \). The second term is a penalty term to fit the diagonal elements of the correlation matrix. The default value is 100. The reciprocal of this value replaces the asymptotic variance corresponding to the diagonal elements of a correlation matrix in the weight matrix \( W \), and it is effective only with the ASYCOV=CORR option. The often used value \( r = 1 \) seems to be too small in many cases to fit the diagonal elements of a correlation matrix properly. The default WPENALTY= value emphasizes the importance of the fit of the diagonal elements in the correlation matrix. You can decrease or increase the value of \( r \) if you want to decrease or increase the importance of the diagonal elements fit. This option is effective only with the WLS or DWLS estimation method and the analysis of a correlation matrix. See the section “Estimation Criteria” on page 490 for more details.

**WRIDGE=r**

defines a ridge factor \( r \) for the diagonal of the weight matrix \( W \) used in GLS, WLS, or DWLS estimation. The weight matrix \( W \) is transformed to

\[
W \rightarrow \tilde{W} = W + r(diag(W))
\]

The WRIDGE= option is applied on the weight matrix

- before the WPENALTY= option is applied on it
- before the weight matrix is written to the OUTWGT= data set
- before the weight matrix is displayed
Since there is no single nonlinear optimization algorithm available that is clearly superior (in terms of stability, speed, and memory) for all applications, different types of optimization techniques are provided in the CALIS procedure. Each technique can be modified in various ways. The default optimization technique for less than 40 parameters ($t < 40$) is TECHNIQUE=LEVMAR. For $40 \leq t < 400$, TECHNIQUE=QUANEW is the default method, and for $t \geq 400$, TECHNIQUE=CONGRA is the default method. For more details, see the section “Use of Optimization Techniques” on page 510. You can specify the following set of options in the PROC CALIS statement or in the NLOPTIONS statement.

**TECHNIQUE | TECH=name**

**OMETHOD | OM=name**

specifies the optimization technique. Valid values for *name* are as follows:

- **CONGRA | CG** chooses one of four different conjugate-gradient optimization algorithms, which can be more precisely defined with the UPDATE= option and modified with the LINESEARCH= option. The conjugate-gradient techniques need only $O(t)$ memory compared to the $O(t^2)$ memory for the other three techniques, where $t$ is the number of parameters. On the other hand, the conjugate-gradient techniques are significantly slower than other optimization techniques and should be used only when insufficient memory is available for more efficient techniques. When you choose this option, UPDATE=PB by default. This is the default optimization technique if there are more than 400 parameters to estimate.

- **DBLDOG | DD** performs a version of double dogleg optimization, which uses the gradient to update an approximation of the Cholesky factor of the Hessian. This technique is, in many aspects, very similar to the dual quasi-Newton method, but it does not use line search. The implementation is based on Dennis and Mei (1979) and Gay (1983).

- **LEVMAR | LM | MARQUARDT** performs a highly stable but, for large problems, memory- and time-consuming Levenberg-Marquardt optimization technique, a slightly improved variant of the Moré (1978) implementation. This is the default optimization technique if there are fewer than 40 parameters to estimate.

- **NEWRAP | NR | NEWTON** performs a usually stable but, for large problems, memory- and time-consuming Newton-Raphson optimization technique. The algorithm combines a line-search algorithm with ridging, and it can be modified with the LINESEARCH= option. In releases prior to Release 6.11, this option invokes the NRRIDG option.

- **NRRIDG | NRR | NR** performs a usually stable but, for large problems, memory- and time-consuming Newton-Raphson optimization technique. This algorithm does not perform a line search. Since
TECH=NRRIDG uses an orthogonal decomposition of the approximate Hessian, each iteration of TECH=NRRIDG can be slower than that of TECH=NEWRAP, which works with Cholesky decomposition. However, usually TECH=NRRIDG needs less iterations than TECH=NEWRAP.

QUANEW | QN chooses one of four different quasi-Newton optimization algorithms that can be more precisely defined with the UPDATE= option and modified with the LINESEARCH= option. If boundary constraints are used, these techniques sometimes converge slowly. When you choose this option, UPDATE=DBFGS by default. If nonlinear constraints are specified in the NLINCON statement, a modification of Powell’s (1982a, 1982b) VMCWD algorithm is used, which is a sequential quadratic programming (SQP) method. This algorithm can be modified by specifying VERSION=1, which replaces the update of the Lagrange multiplier estimate vector \( \mu \) to the original update of Powell (1978a, 1978b) that is used in the VF02AD algorithm. This can be helpful for applications with linearly dependent active constraints. The QUANEW technique is the default optimization technique if there are nonlinear constraints specified or if there are more than 40 and fewer than 400 parameters to estimate. The QUANEW algorithm uses only first-order derivatives of the objective function and, if available, of the nonlinear constraint functions.

TRUREG | TR performs a usually very stable but, for large problems, memory- and time-consuming trust region optimization technique. The algorithm is implemented similar to Gay (1983) and Moré and Sorensen (1983).

NONE | NO does not perform any optimization. This option is similar to METHOD=NONE, but TECH=NONE also computes and displays residuals and goodness-of-fit statistics. If you specify METHOD=ML, METHOD=LSML, METHOD=GLS, METHOD=LSGLS, METHOD=WLS, or METHOD=LSWLS, this option allows computing and displaying (if the display options are specified) of the standard error estimates and modification indices corresponding to the input parameter estimates.

UPDATE | UPD=name specifies the update method for the quasi-Newton or conjugate-gradient optimization technique.

For TECHNIQUE=CONGRA, the following updates can be used:

PB performs the automatic restart update method of Powell (1977) and Beale (1972). This is the default.

FR performs the Fletcher-Reeves update (Fletcher 1980, p. 63).
PR performs the Polak-Ribiere update (Fletcher 1980, p. 66).


For TECHNIQUE=DBLDOG, the following updates (Fletcher 1987) can be used:

DBFGS performs the dual Broyden, Fletcher, Goldfarb, and Shanno (BFGS) update of the Cholesky factor of the Hessian matrix. This is the default.

DDFP performs the dual Davidon, Fletcher, and Powell (DFP) update of the Cholesky factor of the Hessian matrix.

For TECHNIQUE=QUANEW, the following updates (Fletcher 1987) can be used:

BFGS performs original BFGS update of the inverse Hessian matrix. This is the default for earlier releases.

DFP performs the original DFP update of the inverse Hessian matrix.

DBFGS performs the dual BFGS update of the Cholesky factor of the Hessian matrix. This is the default.

DDFP performs the dual DFP update of the Cholesky factor of the Hessian matrix.

LINESearch | LIS | SMETHOD | SM=i
specifies the line-search method for the CONGRA, QUANEW, and NEWRAP optimization techniques. Refer to Fletcher (1980) for an introduction to line-search techniques. The value of i can be 1, . . . , 8; the default is i = 2.

LIS=1 specifies a line-search method that needs the same number of function and gradient calls for cubic interpolation and cubic extrapolation; this method is similar to one used by the Harwell subroutine library.

LIS=2 specifies a line-search method that needs more function calls than gradient calls for quadratic and cubic interpolation and cubic extrapolation; this method is implemented as shown in Fletcher (1987) and can be modified to an exact line search by using the LSPRECISION= option.

LIS=3 specifies a line-search method that needs the same number of function and gradient calls for cubic interpolation and cubic extrapolation; this method is implemented as shown in Fletcher (1987) and can be modified to an exact line search by using the LSPRECISION= option.

LIS=4 specifies a line-search method that needs the same number of function and gradient calls for stepwise extrapolation and cubic interpolation.

LIS=5 specifies a line-search method that is a modified version of LIS=4.

LIS=6 specifies golden section line search (Polak 1971), which uses only function values for linear approximation.
LIS=7 specifies bisection line search (Polak 1971), which uses only function values for linear approximation.

LIS=8 specifies Armijo line-search technique (Polak 1971), which uses only function values for linear approximation.

**FCONV | FTOL=**

specifies the relative function convergence criterion. The optimization process is terminated when the relative difference of the function values of two consecutive iterations is smaller than the specified value of \( r \), that is

\[
\frac{|f(x^{(k)}) - f(x^{(k-1)})|}{\max(|f(x^{(k-1)})|, FSIZE)} \leq r
\]

where \( FSIZE \) can be defined by the FSIZE= option in the NLOPTIONS statement. The default value is \( r = 10^{-FDIGITS} \), where \( FDIGITS \) either can be specified in the NLOPTIONS statement or is set by default to \(-\log_{10}(\epsilon)\), where \( \epsilon \) is the machine precision.

**GCONV | GTOL=**

specifies the relative gradient convergence criterion (see the ABSGCONV= option on page 462 for the absolute gradient convergence criterion).

Termination of all techniques (except the CONGRA technique) requires the normalized predicted function reduction to be small,

\[
\frac{|g(x^{(k)})'^T[G^{(k)}]^{-1}g(x^{(k)})|}{\max(|f(x^{(k)})|, FSIZE)} \leq r
\]

where \( FSIZE \) can be defined by the FSIZE= option in the NLOPTIONS statement. For the CONGRA technique (where a reliable Hessian estimate \( G \) is not available),

\[
\frac{\| g(x^{(k)}) \|_2^2}{\| g(x^{(k)}) - g(x^{(k-1)}) \|_2} \leq r
\]

is used. The default value is \( r = 10^{-8} \).

Note that for releases prior to Release 6.11, the GCONV= option specified the absolute gradient convergence criterion.

**INSTEP=**

For highly nonlinear objective functions, such as the EXP function, the default initial radius of the trust-region algorithms TRUREG, DBLDOG, and LEVMAR or the default step length of the line-search algorithms can produce arithmetic overflows. If this occurs, specify decreasing values of \( 0 < r < 1 \) such as \( \text{INSTEP}=1E-1, \text{INSTEP}=1E-2, \text{INSTEP}=1E-4, \ldots \), until the iteration starts successfully.

- For trust-region algorithms (TRUREG, DBLDOG, and LEVMAR), the INSTEP option specifies a positive factor for the initial radius of the trust region. The default initial trust-region radius is the length of the scaled gradient, and it corresponds to the default radius factor of \( r = 1 \).
• For line-search algorithms (NEWRAP, CONGRA, and QUANEW), INSTEP specifies an upper bound for the initial step length for the line search during the first five iterations. The default initial step length is \( r = 1 \).

For releases prior to Release 6.11, specify the SALPHA= and RADIUS= options. For more details, see the section “Computational Problems” on page 523.

**LSPRECISION | LSP=r**

SPRECISION | SP=r

specifies the degree of accuracy that should be obtained by the line-search algorithms LIS=2 and LIS=3. Usually an imprecise line search is inexpensive and successful. For more difficult optimization problems, a more precise and more expensive line search may be necessary (Fletcher 1980, p.22). The second (default for NEWRAP, QUANEW, and CONGRA) and third line-search methods approach exact line search for small LSPRECISION= values. If you have numerical problems, you should decrease the LSPRECISION= value to obtain a more precise line search. The default LSPRECISION= values are displayed in the following table.

<table>
<thead>
<tr>
<th>TECH=</th>
<th>UPDATE=</th>
<th>LSP default</th>
</tr>
</thead>
<tbody>
<tr>
<td>QUANEW</td>
<td>DBFGS, BFGS</td>
<td>( r = 0.4 )</td>
</tr>
<tr>
<td>QUANEW</td>
<td>DDP, DFP</td>
<td>( r = 0.06 )</td>
</tr>
<tr>
<td>CONGRA</td>
<td>all</td>
<td>( r = 0.1 )</td>
</tr>
<tr>
<td>NEWRAP</td>
<td>no update</td>
<td>( r = 0.9 )</td>
</tr>
</tbody>
</table>

For more details, refer to Fletcher (1980, pp. 25–29).

**MAXFUNC | MAXFU=i**

specifies the maximum number \( i \) of function calls in the optimization process. The default values are displayed in the following table.

<table>
<thead>
<tr>
<th>TECH=</th>
<th>MAXFUNC default</th>
</tr>
</thead>
<tbody>
<tr>
<td>LEVMAR, NEWRAP, NRRIDG, TRUREG</td>
<td>( i = 125 )</td>
</tr>
<tr>
<td>DBLDOG, QUANEW</td>
<td>( i = 500 )</td>
</tr>
<tr>
<td>CONGRA</td>
<td>( i = 1000 )</td>
</tr>
</tbody>
</table>

The default is used if you specify MAXFUNC=0. The optimization can be terminated only after completing a full iteration. Therefore, the number of function calls that is actually performed can exceed the number that is specified by the MAXFUNC= option.

**MAXITER | MAXIT=i <n>**

specifies the maximum number \( i \) of iterations in the optimization process. The default values are displayed in the following table.

<table>
<thead>
<tr>
<th>TECH=</th>
<th>MAXITER default</th>
</tr>
</thead>
<tbody>
<tr>
<td>LEVMAR, NEWRAP, NRRIDG, TRUREG</td>
<td>( i = 50 )</td>
</tr>
<tr>
<td>DBLDOG, QUANEW</td>
<td>( i = 200 )</td>
</tr>
<tr>
<td>CONGRA</td>
<td>( i = 400 )</td>
</tr>
</tbody>
</table>

The default is used if you specify MAXITER=0 or missing.

The optional second value \( n \) is valid only for TECH=QUANEW with nonlinear constraints. It specifies an upper bound \( n \) for the number of iterations of an algorithm.
and reduces the violation of nonlinear constraints at a starting point. The default is \( n = 20 \). For example, specifying

\[
\text{MAXITER} = .0
\]

means that you do not want to exceed the default number of iterations during the main optimization process and that you want to suppress the feasible point algorithm for nonlinear constraints.

\[ \text{RADIUS} = r \]

is an alias for the INSTEP= option for Levenberg-Marquardt minimization.

\[ \text{SALPHA} = r \]

is an alias for the INSTEP= option for line-search algorithms.

\[ \text{SPRECISION} | \text{SP} = r \]

is an alias for the LSPRECISION= option.

**Displayed Output Options**

There are three kinds of options to control the displayed output:

- The PCORR, KURTOSIS, MODIFICATION, NOMOD, PCOVES, PDETERM, PESTIM, PINITIAL, PLATCOV, PREDET, PWEIGHT, RESIDUAL, SIMPLE, STDERR, and TOTEFF options refer to specific parts of displayed output.

- The PALL, PRINT, PSHORT, PSUMMARY, and NOPRINT options refer to special subsets of the displayed output options mentioned in the first item. If the NOPRINT option is not specified, a default set of output is displayed. The PRINT and PALL options add other output options to the default output, and the PSHORT and PSUMMARY options reduce the default displayed output.

- The PRIMAT and PRIVEC options describe the form in which some of the output is displayed (the only nonredundant information displayed by PRIVEC is the gradient).
### Output Options

<table>
<thead>
<tr>
<th>Output Options</th>
<th>PALL</th>
<th>PRINT</th>
<th>default</th>
<th>PSHORT</th>
<th>PSUMMARY</th>
</tr>
</thead>
<tbody>
<tr>
<td>fit indices</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>*</td>
</tr>
<tr>
<td>linear dependencies</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>*</td>
</tr>
<tr>
<td>PREDET</td>
<td>*</td>
<td>(*)</td>
<td>(*)</td>
<td>(*)</td>
<td>*</td>
</tr>
<tr>
<td>model matrices</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>*</td>
</tr>
<tr>
<td>PESTIM</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>*</td>
</tr>
<tr>
<td>iteration history</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>*</td>
</tr>
<tr>
<td>PINITIAL</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>*</td>
</tr>
<tr>
<td>SIMPLE</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>*</td>
</tr>
<tr>
<td>STDERR</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>*</td>
</tr>
<tr>
<td>RESIDUAL</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>*</td>
</tr>
<tr>
<td>KURTOSIS</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>*</td>
</tr>
<tr>
<td>PLATCOV</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>*</td>
</tr>
<tr>
<td>TOTEFF</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>*</td>
</tr>
<tr>
<td>PCORR</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>*</td>
</tr>
<tr>
<td>MODIFICATION</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>*</td>
</tr>
<tr>
<td>PWEIGHT</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>*</td>
</tr>
<tr>
<td>PCOVES</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>*</td>
</tr>
<tr>
<td>PDETERM</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>*</td>
</tr>
<tr>
<td>PRIVEC</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>*</td>
</tr>
</tbody>
</table>

**KURTOSIS | KU**

Computes and displays univariate kurtosis and skewness, various coefficients of multivariate kurtosis, and the numbers of observations that contribute most to the normalized multivariate kurtosis. See the section “Measures of Multivariate Kurtosis” on page 503 for more information. Using the KURTOSIS option implies the SIMPLE display option. This information is computed only if the DATA= data set is a raw data set, and it is displayed by default if the PRINT option is specified. The multivariate LS kappa and the multivariate mean kappa are displayed only if you specify METHOD=WLS and the weight matrix is computed from an input raw data set. All measures of skewness and kurtosis are corrected for the mean. If an intercept variable is included in the analysis, the measures of multivariate kurtosis do not include the intercept variable in the corrected covariance matrix, as indicated by a displayed message. Using the BIASKUR option displays the biased values of univariate skewness and kurtosis.

**MODIFICATION | MOD**

Computes and displays Lagrange multiplier test indices for constant parameter constraints, equality parameter constraints, and active boundary constraints, as well as univariate and multivariate Wald test indices. The modification indices are not computed in the case of unweighted or diagonally weighted least-squares estimation.

The Lagrange multiplier test (Bentler 1986; Lee 1985; Buse 1982) provides an estimate of the $\chi^2$ reduction that results from dropping the constraint. For constant parameter constraints and active boundary constraints, the approximate change of the parameter value is displayed also. You can use this value to obtain an initial value if the parameter is allowed to vary in a modified model. For more information, see the section “Modification Indices” on page 519.
NOMOD

does not compute modification indices. The NOMOD option is useful in connection with the PALL option because it saves computing time.

NOPRINT | NOP

suppresses the displayed output. Note that this option temporarily disables the Output Delivery System (ODS). For more information, see Chapter 14, “Using the Output Delivery System.”

PALL | ALL

displays all optional output except the output generated by the PCOVES option, the output generated by the PDETERM option, and the output generated by the PRIVEC option.

CAUTION: The PALL option includes the very expensive computation of the modification indices. If you do not really need modification indices, you can save computing time by specifying the NOMOD option in addition to the PALL option.

PCORR | CORR

displays the (corrected or uncorrected) covariance or correlation matrix that is analyzed and the predicted model covariance or correlation matrix.

PCOVES | PCE

displays the following:

- the information matrix (crossproduct Jacobian)
- the approximate covariance matrix of the parameter estimates (generalized inverse of the information matrix)
- the approximate correlation matrix of the parameter estimates

The covariance matrix of the parameter estimates is not computed for estimation methods ULS and DWLS. This displayed output is not included in the output generated by the PALL option.

PDETERM | PDE

displays three coefficients of determination: the determination of all equations (DETAE), the determination of the structural equations (DETSE), and the determination of the manifest variable equations (DETMV). These determination coefficients are intended to be global means of the squared multiple correlations for different subsets of model equations and variables. The coefficients are displayed only when you specify a RAM or LINEQS model, but they are displayed for all five estimation methods: ULS, GLS, ML, WLS, and DWLS.

You can use the STRUCTEQ statement to define which equations are structural equations. If you don’t use the STRUCTEQ statement, PROC CALIS uses its own default definition to identify structural equations.

The term “structural equation” is not defined in a unique way. The LISREL program defines the structural equations by the user-defined BETA matrix. In PROC CALIS, the default definition of a structural equation is an equation that has a dependent left side variable that appears at least once on the right side of another equation, or an equation that has at least one right side variable that is the left side variable of another
PROC CALIS Statement

Therefore, PROC CALIS sometimes identifies more equations as structural equations than the LISREL program does.

If the model contains structural equations, PROC CALIS also displays the “Stability Coefficient of Reciprocal Causation,” that is, the largest eigenvalue of the $BB'$ matrix, where $B$ is the causal coefficient matrix of the structural equations. These coefficients are computed as in the LISREL VI program of Jöreskog and Sörbom (1985). This displayed output is not included in the output generated by the PALL option.

PESTIM | PES
displays the parameter estimates. In some cases, this includes displaying the standard errors and $t$ values.

PINITIAL | PIN
displays the input model matrices and the vector of initial values.

PJACPAT | PJP
displays the structure of variable and constant elements of the Jacobian matrix.

PLATCOV | PLC
displays the following:

- the estimates of the covariances among the latent variables
- the estimates of the covariances between latent and manifest variables
- the latent variable score regression coefficients

The estimated covariances between latent and manifest variables and the latent variable score regression coefficients are written to the OUTSTAT= data set. You can use the score coefficients with PROC SCORE to compute factor scores.

PREDET | PRE
displays the pattern of variable and constant elements of the predicted moment matrix that is predetermined by the analysis model. It is especially helpful in finding manifest variables that are not used or that are used as exogenous variables in a complex model specified in the COSAN statement. Those entries of the predicted moment matrix for which the model generates variable (rather than constant) elements are displayed as missing values. This output is displayed even without specifying the PREDET option if the model generates constant elements in the predicted model matrix different from those in the analysis moment matrix and if you specify at least the PSHORT amount of displayed output.

If the analyzed matrix is a correlation matrix (containing constant elements of 1s in the diagonal) and the model generates a predicted model matrix with $q$ constant (rather than variable) elements in the diagonal, the degrees of freedom are automatically reduced by $q$. The output generated by the PREDET option displays those constant diagonal positions. If you specify the DFREDUCE= or NODIAG option, this automatic reduction of the degrees of freedom is suppressed. See the section “Counting the Degrees of Freedom” on page 522 for more information.
PRIMAT | PMAT
---
displays parameter estimates, approximate standard errors, and \( t \) values in matrix form if you specify the analysis model in the RAM or LINEQS statement. When a COSAN statement is used, this occurs by default.

PRINT | PRI
---
adds the options KURTOSIS, RESIDUAL, PLATCOV, and TOTTEFF to the default output.

PRIVEC | PVEC
---
displays parameter estimates, approximate standard errors, the gradient, and \( t \) values in vector form. The values are displayed with more decimal places. This displayed output is not included in the output generated by the PALL option.

PSHORT | SHORT | PSH
---
excludes the output produced by the PINITIAL, SIMPLE, and STDERR options from the default output.

PSUMMARY | SUMMARY | PSUM
---
displays the fit assessment table and the ERROR, WARNING, and NOTE messages.

PWEIGHT | PW
---
displays the weight matrix \( W \) used in the estimation. The weight matrix is displayed after the WRIDGE= and the WPENALTY= options are applied to it.

RESIDUAL | RES < = NORM | VARSTAND | ASYSTAND >
---
displays the absolute and normalized residual covariance matrix, the rank order of the largest residuals, and a bar chart of the residuals. This information is displayed by default when you specify the PRINT option. Three types of normalized or standardized residual matrices can be chosen with the RESIDUAL= specification:

- RESIDUAL= NORM  Normalized Residuals
- RESIDUAL= VARSTAND  Variance Standardized Residuals
- RESIDUAL= ASYSTAND  Asymptotically Standardized Residuals

For more details, see the section “Assessment of Fit” on page 495.

SIMPLE | S
---
displays means, standard deviations, skewness, and univariate kurtosis if available. This information is displayed when you specify the PRINT option. If you specify the UCOV, UCORR, or NOINT option, the standard deviations are not corrected for the mean. If the KURTOSIS option is specified, the SIMPLE option is set by default.

STDERR | SE
---
displays approximate standard errors if estimation methods other than unweighted least squares (ULS) or diagonally weighted least squares (DWLS) are used (and the NOSTDERR option is not specified). If you specify neither the STDERR nor the NOSTDERR option, the standard errors are computed for the OUTRAM= data set. This information is displayed by default when you specify the PRINT option.

TOTTEFF | TE
---
computes and displays total effects and indirect effects.
Miscellaneous Options

ALPHAECV=\alpha
specifies the significance level for a \(1 - \alpha\) confidence interval, \(0 \leq \alpha \leq 1\), for the Browne & Cudeck (1993) expected cross validation index (ECVI). The default value is \(\alpha = 0.1\), which corresponds to a 90% confidence interval for the ECVI.

ALPHARMS=\alpha
specifies the significance level for a \(1 - \alpha\) confidence interval, \(0 \leq \alpha \leq 1\), for the Steiger & Lind (1980) root mean squared error of approximation (RMSEA) coefficient (refer to Browne and Du Toit 1992). The default value is \(\alpha = 0.1\), which corresponds to a 90% confidence interval for the RMSEA.

ASINGULAR | ASING=r
specifies an absolute singularity criterion \(r, r > 0\), for the inversion of the information matrix, which is needed to compute the covariance matrix. The following singularity criterion is used:

\[ |d_{j,j}| \leq \max(ASING, VSING * |H_{j,j}|, MSING * \max(|H_{1,1}|, \ldots, |H_{n,n}|)) \]

where \(d_{j,j}\) is the diagonal pivot of the matrix, and \(VSING\) and \(MSING\) are the specified values of the VSINGULAR= and MSINGULAR= options. The default value for ASING is the square root of the smallest positive double precision value. Note that, in many cases, a normalized matrix \(D^{-1}HD^{-1}\) is decomposed, and the singularity criteria are modified correspondingly.

BIASKUR
computes univariate skewness and kurtosis by formulas uncorrected for bias. See the section “Measures of Multivariate Kurtosis” on page 503 for more information.

COVSING=r
specifies a nonnegative threshold \(r\), which determines whether the eigenvalues of the information matrix are considered to be zero. If the inverse of the information matrix is found to be singular (depending on the VSINGULAR=, MSINGULAR=, ASINGULAR=, or SINGULAR= option), a generalized inverse is computed using the eigenvalue decomposition of the singular matrix. Those eigenvalues smaller than \(r\) are considered to be zero. If a generalized inverse is computed and you do not specify the NOPRINT option, the distribution of eigenvalues is displayed.

DEMPHAS | DE=r
changes the initial values of all parameters that are located on the diagonals of the central model matrices by the relationship

\[ \text{diag}_{\text{new}} = r(\text{diag}_{\text{old}} + 1) \]

The initial values of the diagonal elements of the central matrices should always be nonnegative to generate positive definite predicted model matrices in the first iteration. By using values of \(r > 1\), for example, \(r = 2, r = 10, \ldots\), you can increase these initial values to produce predicted model matrices with high positive eigenvalues in the first iteration. The DEMPHAS= option is effective independent of the way
the initial values are set; that is, it changes the initial values set in the model specification as well as those set by an INRAM= data set and those automatically generated for RAM, LINEQS, or FACTOR model statements. It also affects the initial values set by the START= option, which uses, by default, DEMPHAS=100 if a covariance matrix is analyzed and DEMPHAS=10 for a correlation matrix.

**FDCODE**

replaces the analytic derivatives of the program statements by numeric derivatives (finite difference approximations). In general, this option is needed only when you have program statements that are too difficult for the built-in function compiler to differentiate analytically. For example, if the program code for the nonlinear constraints contains many arrays and many DO loops with array processing, the built-in function compiler can require too much time and memory to compute derivatives of the constraints with respect to the parameters. In this case, the Jacobian matrix of constraints is computed numerically by using finite difference approximations. The FDCODE option does not modify the kind of derivatives specified with the HESSALG= option.

**HESSALG | HA = 1 | 2 | 3 | 4 | 5 | 6 | 11**

specifies the algorithm used to compute the (approximate) Hessian matrix when TECHNIQUE=LEVMAR and NEWRAP, to compute approximate standard errors of the parameter estimates, and to compute Lagrange multipliers. There are different groups of algorithms available:

- analytic formulas: HA=1,2,3,4,11
- finite difference approximation: HA=5,6
- dense storage: HA=1,2,3,4,5,6
- sparse storage: HA=11

If the Jacobian is more than 25% dense, the dense analytic algorithm, HA= 1, is used by default. The HA= 1 algorithm is faster than the other dense algorithms, but it needs considerably more memory for large problems than HA= 2,3,4. If the Jacobian is more than 75% sparse, the sparse analytic algorithm, HA= 11, is used by default. The dense analytic algorithm HA= 4 corresponds to the original COSAN algorithm; you are advised not to specify HA= 4 due to its very slow performance. If there is not enough memory available for the dense analytic algorithm HA= 1 and you must specify HA= 2 or HA= 3, it may be more efficient to use one of the quasi-Newton or conjugate-gradient optimization techniques since Levenberg-Marquardt and Newton-Raphson optimization techniques need to compute the Hessian matrix in each iteration. For approximate standard errors and modification indices, the Hessian matrix has to be computed at least once, regardless of the optimization technique.

The algorithms HA= 5 and HA= 6 compute approximate derivatives by using forward difference formulas. The HA= 5 algorithm corresponds to the analytic HA= 1: it is faster than HA= 6, however it needs much more memory. The HA= 6 algorithm corresponds to the analytic HA= 2: it is slower than HA= 5, however it needs much less memory.
Test computations of large sparse problems show that the sparse algorithm HA= 11 can be up to ten times faster than HA= 1 (and needs much less memory).

**MSINGULAR | MSING=r**
specifies a relative singularity criterion \( r, r > 0 \), for the inversion of the information matrix, which is needed to compute the covariance matrix. The following singularity criterion is used:

\[
|d_{j,j}| \leq \max(\text{ASING}, \text{VSING} \times |H_{j,j}|, \text{MSING} \times \max(|H_{1,1}|, \ldots, |H_{n,n}|))
\]

where \( d_{j,j} \) is the diagonal pivot of the matrix, and ASING and VSING are the specified values of the ASINGULAR= and VSINGULAR= options. If you do not specify the SINGULAR= option, the default value for \( \text{MSING} \) is \( 1E^{-12} \); otherwise, the default value is \( 1E^{-4} \times \text{SINGULAR} \). Note that, in many cases, a normalized matrix \( D^{-1}HD^{-1} \) is decomposed, and the singularity criteria are modified correspondingly.

**NOSTDERR | NOSE**
specifies that standard errors should not be computed. Standard errors are not computed for unweighted least-squares (ULS) or diagonally weighted least-squares (DWLS) estimation. In general, standard errors are computed even if the STDERR display option is not used (for file output).

**RANDOM = i**
specifies a positive integer as a seed value for the pseudo-random number generator to generate initial values for the parameter estimates for which no other initial value assignments in the model definitions are made. Except for the parameters in the diagonal locations of the central matrices in the model, the initial values are set to random numbers in the range \( 0 \leq r \leq 1 \). The values for parameters in the diagonals of the central matrices are random numbers multiplied by 10 or 100. For more information, see the section “Initial Estimates” on page 506.

**SINGULAR | SING =r**
specifies the singularity criterion \( r, 0 < r < 1 \), used, for example, for matrix inversion. The default value is the square root of the relative machine precision or, equivalently, the square root of the largest double precision value that, when added to 1, results in 1.

**SLMW=r**
specifies the probability limit used for computing the stepwise multivariate Wald test. The process stops when the univariate probability is smaller than \( r \). The default value is \( r = 0.05 \).

**START =r**
In general, this option is needed only in connection with the COSAN model statement, and it specifies a constant \( r \) as an initial value for all the parameter estimates for which no other initial value assignments in the pattern definitions are made. Start values in the diagonal locations of the central matrices are set to \( 100|r| \) if a COV or UCOV matrix is analyzed and \( 10|r| \) if a CORR or UCORR matrix is analyzed. The default value is \( r = .5 \). Unspecified initial values in a FACTOR, RAM, or LINEQS model are usually computed by PROC CALIS. If none of the initialization methods...
are able to compute all starting values for a model specified by a FACTOR, RAM, or LINEQS statement, then the start values of parameters that could not be computed are set to \( r, 10^{|r|}, \) or \( 100^{|r|}. \) If the DEMPHAS= option is used, the initial values of the diagonal elements of the central model matrices are multiplied by the value specified in the DEMPHAS= option. For more information, see the section “Initial Estimates” on page 506.

**VSINGULAR | VSING=r**

specifies a relative singularity criterion \( r, r > 0, \) for the inversion of the information matrix, which is needed to compute the covariance matrix. The following singularity criterion is used:

\[
|d_{j,j}| \leq \max(ASING, VSING \times |H_{j,j}|, MSING \times \max(|H_{1,1}|, \ldots, |H_{n,n}|))
\]

where \( d_{j,j} \) is the diagonal pivot of the matrix, and ASING and MSING are the specified values of the ASINGULAR= and MSINGULAR= options. If you do not specify the SINGULAR= option, the default value for VSING is 1E−8; otherwise, the default value is SINGULAR. Note that in many cases a normalized matrix \( D^{-1}HD^{-1} \) is decomposed, and the singularity criteria are modified correspondingly.

---

**COSAN Model Statement**

```plaintext
COSAN matrix_term < + matrix_term ... > ;
where matrix_term represents

matrix_definition < * matrix_definition ... >
and matrix_definition represents

matrix_name (column_number < , general_form < , transformation > )
```

The COSAN statement constructs the symmetric matrix model for the covariance analysis mentioned earlier (see the section “The Generalized COSAN Model” on page 400):

\[
C = F_1P_1F_1' + \cdots + F_mP_mF_m',
\]

\[
F_k = F_k1 \cdots F_k_{n(k)}, \quad \text{and} \quad P_k = P_k', \quad k = 1, \ldots, m
\]

\[
F_{kj} = \begin{cases} 
G_{kj} & j = 1, \ldots, n(k), \\
G_{kj}^{-1} & \\
(I - G_{kj})^{-1} & 
\end{cases} \quad \text{and} \quad P_k = \begin{cases} 
Q_k \\
Q_k^{-1} 
\end{cases}
\]

You can specify only one COSAN statement with each PROC CALIS statement. The COSAN statement contains \( m \) matrix_terms corresponding to the generalized COSAN formula. The matrix_terms are separated by plus signs (+) according to the addition of the terms within the model.

Each matrix_term of the COSAN statement contains the definitions of the first \( n(k) + 1 \) matrices, \( F_{kj} \) and \( P_k \), separated by asterisks (*) according to the multiplication of
the matrices within the term. The matrices $F'_{ik}$ of the right-hand-side product are redundant and are not specified within the COSAN statement.

Each matrix definition consists of the name of the matrix (matrix_name), followed in parentheses by the number of columns of the matrix (column_number) and, optionally, one or two matrix properties, separated by commas, describing the form of the matrix.

The number of rows of the first matrix in each term is defined by the input correlation or covariance matrix. You can reorder and reduce the variables in the input moment matrix using the VAR statement. The number of rows of the other matrices within the term is defined by the number of columns of the preceding matrix.

The first matrix property describes the general form of the matrix in the model. You can choose one of the following specifications of the first matrix property. The default first matrix property is GEN.

<table>
<thead>
<tr>
<th>Code</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>IDE</td>
<td>specifies an identity matrix; if the matrix is not square, this specification describes an identity submatrix followed by a rectangular zero submatrix.</td>
</tr>
<tr>
<td>ZID</td>
<td>specifies an identity matrix; if the matrix is not square, this specification describes a rectangular zero submatrix followed by an identity submatrix.</td>
</tr>
<tr>
<td>DIA</td>
<td>specifies a diagonal matrix; if the matrix is not square, this specification describes a diagonal submatrix followed by a rectangular zero submatrix.</td>
</tr>
<tr>
<td>ZDI</td>
<td>specifies a diagonal matrix; if the matrix is not square, this specification describes a rectangular zero submatrix followed by a diagonal submatrix.</td>
</tr>
<tr>
<td>LOW</td>
<td>specifies a lower triangular matrix; the matrix can be rectangular.</td>
</tr>
<tr>
<td>UPP</td>
<td>specifies an upper triangular matrix; the matrix can be rectangular.</td>
</tr>
<tr>
<td>SYM</td>
<td>specifies a symmetric matrix; the matrix cannot be rectangular.</td>
</tr>
<tr>
<td>GEN</td>
<td>specifies a general rectangular matrix (default).</td>
</tr>
</tbody>
</table>

The second matrix property describes the kind of inverse matrix transformation. If the second matrix property is omitted, no transformation is applied to the matrix.

<table>
<thead>
<tr>
<th>Code</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>INV</td>
<td>uses the inverse of the matrix.</td>
</tr>
<tr>
<td>IMI</td>
<td>uses the inverse of the difference between the identity and the matrix.</td>
</tr>
</tbody>
</table>

You cannot specify a nonsquare parameter matrix as an INV or IMI model matrix. Specifying a matrix of type DIA, ZDI, UPP, LOW, or GEN is not necessary if you do not use the unspecified location list in the corresponding MATRIX statements. After PROC CALIS processes the corresponding MATRIX statements, the matrix type DIA, ZDI, UPP, LOW, or GEN is recognized from the pattern of possibly nonzero elements. If you do not specify the first matrix property and you use the unspecified location list in a corresponding MATRIX statement, the matrix is recognized as a
GEN matrix. You can also generate an IDE or ZID matrix by specifying a DIA, ZDI, or IMI matrix and by using MATRIX statements that define the pattern structure. However, PROC CALIS would be unable to take advantage of the fast algorithms that are available for IDE and ZID matrices in this case.

For example, to specify a second-order factor analysis model

\[
S = F_1F_2P_2F_1'F_1' + F_1U_2^3F_1' + U_1^2
\]

with \( m_1 = 3 \) first-order factors and \( m_2 = 2 \) second-order factors and with \( n = 9 \) variables, you can use the following COSAN statement:

\[
\text{COSAN } F1(3) * F2(2) * P2(2,SYM) + F1(3) * U2(3,DIA) * I1(3,IDE) \\
+ U1(9,DIA) * I2(9,IDE)
\]

**MATRIX Statement**

\[
\text{MATRIX } \text{matrix-name} < \text{location} > = \text{list}, \text{location} = \text{list} \ldots ;
\]

You can specify one or more MATRIX statements with a COSAN or FACTOR statement. A MATRIX statement specifies which elements of the matrix are constant and which are parameters. You can also assign values to the constant elements and initial values for the parameters. The input notation resembles that used in the COSAN program of R. McDonald and C. Fraser (personal communication), except that in PROC CALIS, parameters are distinguished from constants by giving parameters names instead of by using positive and negative integers.

A MATRIX statement cannot be used for an IDE or ZID matrix. For all other types of matrices, each element is assumed to be a constant of 0 unless a MATRIX statement specifies otherwise. Hence, there must be at least one MATRIX statement for each matrix mentioned in the COSAN statement except for IDE and ZID matrices. There can be more than one MATRIX statement for a given matrix. If the same matrix element is given different definitions, later definitions override earlier definitions.

At the start, all elements of each model matrix, except IDE or ZID matrices, are set equal to 0.

**Description of location:**

There are several ways to specify the starting location and continuation direction of a list with \( n + 1, n \geq 0 \), elements within the parameter matrix.

\[[i,j]\] The list elements correspond to the diagonally continued matrix elements \([i,j], [i+1,j+1], \ldots, [i+n,j+n]\). The number of elements is defined by the length of the list and eventually terminated by the matrix boundaries. If the list contains just one element (constant or variable), then it is assigned to the matrix element \([i,j]\).
The list elements correspond to the horizontally continued matrix elements \([i,j], [i,j+1], \ldots, [i,j+n]\), where the starting column \(j\) is the diagonal position for a DIA, ZDI, or UPP matrix and is the first column for all other matrix types. For a SYM matrix, the list elements refer only to the matrix elements in the lower triangle. For a DIA or ZDI matrix, only one list element is accepted.

The list elements correspond to the vertically continued matrix elements \([i,j], [i+1,j], \ldots, [i+n,j]\), where the starting row \(i\) is equal to the diagonal position for a DIA, ZDI, SYM, or LOW matrix and is the first row for each other matrix type. For a SYM matrix, the list elements refer only to the matrix elements in the lower triangle. For a DIA or ZDI matrix, only one list element is accepted.

unspecified location: The list is allocated to all valid matrix positions (except for a ZDI matrix) starting at the element \([1,1]\) and continuing rowwise. The only valid matrix positions for a DIA or ZDI matrix are the diagonal elements; for an UPP or LOW matrix, the valid positions are the elements above or below the diagonal; and for a symmetric matrix, the valid positions are the elements in the lower triangle since the other triangle receives the symmetric allocation automatically. This location definition differs from the definitions with specified pattern locations in one important respect: if the number of elements in the list is smaller than the number of valid matrix elements, the list is repeated in the allocation process until all valid matrix elements are filled.

Omitting the left-hand-side term is equivalent to using \([,]\) for an unspecified location.

Description of list:

The list contains numeric values or parameter names, or both, that are assigned to a list of matrix elements starting at a specified position and proceeding in a specified direction. A real number \(r\) in the list defines the corresponding matrix element as a constant element with this value. The notation \(n * r\) generates \(n\) values of \(r\) in the list. A name in the list defines the corresponding matrix element as a parameter to be estimated. You can use numbered name lists (X1-X10) or the asterisk notation (5 *X means five occurrences of the parameter X). If a sublist of \(n_1\) names inside a list is followed by a list of \(n_2 \leq n_1\) real values inside parentheses, the last \(n_2\) parameters in the name sublist are given the initial values mentioned inside the parenthesis. For example, the following list

\[
0. 1. A2-A5 (1.4 1.9 2.5) 5.
\]

specifies that the first two matrix elements (specified by the location to the left of the equal sign) are constants with values 0 and 1. The next element is parameter A2 with no specified initial value. The next three matrix elements are the variable parameters A3, A4, and A5 with initial values 1.4, 1.9, and 2.5, respectively. The next matrix element is specified by the seventh list element to be the constant 5.

If your model contains many unconstrained parameters and it is too cumbersome to find different parameter names, you can specify all those parameters by the same
prefix name. A prefix is a short name followed by a colon. The CALIS procedure
generates a parameter name by appending an integer suffix to this prefix name. The
prefix name should have no more than five or six characters so that the generated
parameter name is not longer than eight characters. For example, if the prefix A (the
parameter A1) is already used once in a list, the previous example would be identical
to

0.1.4*A: (1.4 1.9 2.5) 5.

To avoid unintentional equality constraints, the prefix names should not coincide with
explicitly defined parameter names.

If you do not assign initial values to the parameters (listed in parentheses following a
name sublist within the pattern list), PROC CALIS assigns initial values as follows:

- If the PROC CALIS statement contains a START=r option, each uninitialized
  parameter is given the initial value r. The uninitialized parameters in the diagonals
  of the central model matrices are given the initial value 10|r|, 100|r|, or |r| multiplied
  by the value specified in the DEMPHAS= option.
- If the PROC CALIS statement contains a RANDOM=i option, each uninitia-
  lized parameter is given a random initial value 0 ≤ r ≤ 1. The uninitialized
  parameters in the diagonals of the central model matrices are given the random
  values multiplied by 10, 100, or the value specified in the DEMPHAS= option.
- Otherwise, the initial value is set corresponding to START=0.5.

For example, to specify a confirmatory second-order factor analysis model

\[ S = F_1F_2P_2F_2'F_1' + F_1U_2^2F_1' + U_1^2 \]

with \( m_1 = 3 \) first-order factors, \( m_2 = 2 \) second-order factors, and \( n = 9 \) variables
and the following matrix pattern,

\[
F_1 = \begin{pmatrix}
X_1 & 0 & 0 \\
X_2 & 0 & 0 \\
X_3 & 0 & 0 \\
0 & X_4 & 0 \\
0 & X_5 & 0 \\
0 & X_6 & 0 \\
0 & 0 & X_7 \\
0 & 0 & X_8 \\
0 & 0 & X_9
\end{pmatrix}, \quad U_1 = \begin{pmatrix}
U_1 \\
U_2 \\
U_3 \\
U_4 \\
U_5 \\
U_6 \\
U_7 \\
U_8 \\
U_9
\end{pmatrix}
\]

\[
F_2 = \begin{pmatrix}
Y_1 & 0 \\
Y_1 & Y_2 \\
0 & Y_2
\end{pmatrix}, \quad P_2 = \begin{pmatrix}
P & 0 \\
0 & P
\end{pmatrix}, \quad U_2 = \begin{pmatrix}
V_1 \\
V_2 \\
V_3
\end{pmatrix}
\]

you can specify the following COSAN and MATRIX statements:
cosan f1(3) * f2(2) * p2(2,dia) + f1(3) * u2(3,dia) * i1(3,ide) + u1(9,dia) * i2(9,ide);

matrix f1
   [ ,1] = x1-x3,
   [ ,2] = 3 * 0 x4-x6,
   [ ,3] = 6 * 0 x7-x9;

matrix u1
   [1,1] = u1-u9;

matrix f2
   [ ,1] = 2 * y1,
   [ ,2] = 0.2 * y2;

matrix u2 = 3 * v;

matrix p2 = 2 * p;
run;

The matrix pattern includes several equality constraints. Two loadings in the first and second factor of \(F_2\) (parameter names \(Y_1\) and \(Y_2\)) and the two factor correlations in the diagonal of matrix \(P_2\) (parameter name \(P\)) are constrained to be equal. There are many other ways to specify the same model. See Figure 17.2 for the path diagram of this model.

**Using the FACTOR and MATRIX Statements**

You can specify the MATRIX statement and the FACTOR statement to compute a confirmatory first-order factor or component analysis. You can define the elements of the matrices \(F\), \(P\), and \(U\) of the oblique model,

\[
C = FP\hat{F}' + U^2, \quad P = P', \quad U = \text{diag}
\]

To specify the structure for matrix \(F\), \(P\), or \(U\), you have to refer to the matrix \_F_, \_P_, or \_U_ in the MATRIX statement. (Matrix names automatically set by PROC CALIS always start with an underscore. As you name your own matrices or variables, you should avoid leading underscores.)

The default matrix forms are as follows.

- \(F\) lower triangular matrix (0 upper triangle for problem identification, removing rotational invariance)
- \(P\) identity matrix (constant)
- \(U\) diagonal matrix

If you are using at least one MATRIX statement in connection with a FACTOR model statement, you can also use the BOUNDS or PARAMETERS statement and program statements to constrain the parameters named in the MATRIX statement. Initial estimates are computed by McDonald’s (McDonald and Hartmann 1992) method. McDonald’s method of computing initial values works better if you scale the factors by setting the factor variances to 1 rather than by setting the loadings of the reference variables equal to 1.
The RAM statement defines the elements of the symmetric RAM matrix model
\[ \mathbf{v} = \mathbf{A}\mathbf{v} + \mathbf{u} \]
in the form of a list type input (McArdle and McDonald 1984).

The covariance structure is given by
\[ \mathbf{C} = \mathbf{J}(\mathbf{I} - \mathbf{A})^{-1}\mathbf{P}(\mathbf{I} - \mathbf{A})^{-1}'\mathbf{J}' \]

with selection matrix \( \mathbf{J} \) and
\[ \mathbf{C} = \mathcal{E}\{\mathbf{J}\mathbf{vv}'\mathbf{J}'\}, \quad \mathbf{P} = \mathcal{E}\{\mathbf{uu}'\} \]

You can specify only one RAM statement with each PROC CALIS statement. Using the RAM statement requires that the first \( n \) variable numbers in the path diagram and in the vector \( \mathbf{v} \) correspond to the numbers of the \( n \) manifest variables of the given covariance or correlation matrix. If you are not sure what the order of the manifest variables in the DATA= data set is, use a VAR statement to specify the order of these observed variables. Using the AUGMENT option includes the INTERCEPT variable as a manifest variable with number \( n + 1 \) in the RAM model. In this case, latent variables have to start with \( n + 2 \). The box of each manifest variable in the path diagram is assigned the number of the variable in the covariance or correlation matrix.

The selection matrix \( \mathbf{J} \) is always a rectangular identity (IDE) matrix, and it does not have to be specified in the RAM statement. A constant matrix element is defined in a RAM statement by a \texttt{list-entry} with four numbers. You define a parameter element by three or four numbers followed by a name for the parameter. Separate the list entries with a comma. Each \texttt{list-entry} in the RAM statement corresponds to a path in the diagram, as follows:

- The first number in each list entry (\texttt{matrix-number}) is the number of arrow heads of the path, which is the same as the number of the matrix in the RAM model (1 := \texttt{A} , 2 := \texttt{P}).
- The second number in each list entry (\texttt{row-number}) is the number of the node in the diagram to which the path points, which is the same as the row number of the matrix element.
- The third number in each list entry (\texttt{column-number}) is the number of the node in the diagram from which the path originates, which is the same as the column number of the matrix element.
The fourth number (value) gives the (initial) value of the path coefficient. If you do not specify a fifth list-entry, this number specifies a constant coefficient; otherwise, this number specifies the initial value of this parameter. It is not necessary to specify the fourth item. If you specify neither the fourth nor the fifth item, the constant is set to 1 by default. If the fourth item (value) is not specified for a parameter, PROC CALIS tries to compute an initial value for this parameter.

If the path coefficient is a parameter rather than a constant, then a fifth item in the list entry (parameter-name) is required to assign a name to the parameter. Using the same name for different paths constrains the corresponding coefficients to be equal.

If the initial value of a parameter is not specified in the list, the initial value is chosen in one of the following ways:

- If the PROC CALIS statement contains a RANDOM=i option, then the parameter obtains a randomly generated initial value \( r \), such that \( 0 \leq r \leq 1 \). The uninitialized parameters in the diagonals of the central model matrices are given the random values \( r \) multiplied by 10, 100, or the value specified in the DEPHAS= option.
- If the RANDOM= option is not used, PROC CALIS tries to estimate the initial values.
- If the initial values cannot be estimated, the value of the START= option is used as an initial value.

If your model contains many unconstrained parameters and it is too cumbersome to find different parameter names, you can specify all those parameters by the same prefix name. A prefix is a short name followed by a colon. The CALIS procedure then generates a parameter name by appending an integer suffix to this prefix name. The prefix name should have no more than five or six characters so that the generated parameter name is not longer than eight characters. To avoid unintentional equality constraints, the prefix names should not coincide with explicitly defined parameter names.

For example, you can specify the confirmatory second-order factor analysis model (mentioned on page 441)

\[
S = F_1F_2'P_2F_2F_1' + F_1U_2^2F_1' + U_1^2
\]

using the following RAM model statement:

```
ram
  1  1  10  x1,
  1  2  10  x2,
  1  3  10  x3,
  1  4  11  x4,
  1  5  11  x5,
  1  6  11  x6,
  1  7  12  x7,
```
The confirmatory second-order factor analysis model corresponds to the path diagram displayed in Figure 17.2.

Figure 17.2. Path Diagram of Second-Order Factor Analysis Model

There is a very close relationship between the RAM model algebra and the specification of structural linear models by path diagrams. See Figure 17.3 for an example.
1. Multiple Regression

2. Chain Simplex

3. First-Order Factor Analysis

4. Second-Order Factor Analysis

**Figure 17.3.** Examples of RAM Nomography

Refer to McArdle (1980) for the interpretation of the models displayed in Figure 17.3.

**LINEQS Model Statement**

LINEQS  *equation < , equation ... > ;*
where *equation* represents dependent = term < + term ... >
and where *term* represents one of the following:

- coefficient-name < (number) > variable-name
- prefix-name < (number) > variable-name
- < number > variable-name
The LINEQS statement defines the LINEQS model

\[
\eta = \beta \eta + \gamma \xi \\
C = J(I - B)^{-1} \Phi \Phi'((I - B)^{-1})'J'
\]

You can specify only one LINEQS statement with each PROC CALIS statement. There are some differences from Bentler's notation in choosing the variable names. The length of each variable name is restricted to eight characters. The names of the manifest variables are defined in the DATA= input data set. The VAR statement can be used to select a subset of manifest variables in the DATA= input data set to analyze. You do not need to use a V prefix for manifest variables in the LINEQS statement nor do you need to use a numerical suffix in any variable name. The names of the latent variables must start with the prefix letter F (for Factor); the names of the residuals must start with the prefix letters E (for Error) or D (for Disturbance). The trailing part of the variable name can contain letters or digits. The prefix letter E is used for the errors of the manifest variables, and the prefix letter D is used for the disturbances of the latent variables. The names of the manifest variables in the DATA= input data set can start with F, E, or D, but these names should not coincide with the names of latent or error variables used in the model. The left-hand side (that is, endogenous dependent variable) of each equation should be either a manifest variable of the data set or a latent variable with prefix letter F. The left-hand-side variable should not appear on the right-hand side of the same equation; this means that matrix should not have a nonzero diagonal element. Each equation should contain, at most, one E or D variable.

The equations must be separated by a comma. The order of the equations is arbitrary. The displayed output generally contains equations and terms in an order different from the input.

Coefficients to estimate are indicated in the equations by a name preceding the independent variable’s name. The coefficient’s name can be followed by a number inside parentheses indicating the initial value for this coefficient. A number preceding the independent variable’s name indicates a constant coefficient. If neither a coefficient name nor a number precedes the independent variable’s name, a constant coefficient of 1 is assumed.

If the initial value of a parameter is not specified in the equation, the initial value is chosen in one of the following ways:

- If you specify the RANDOM= option in the PROC CALIS statement, the variable obtains a randomly generated initial value \( r \), such that \( 0 \leq r \leq 1 \). The uninitialized parameters in the diagonals of the central model matrices are given the nonnegative random values \( r \) multiplied by 10, 100, or the value specified in the DEPHAS= option.
- If the RANDOM= option is not used, PROC CALIS tries to estimate the initial values.
If the initial values cannot be estimated, the value of the START= option is used as an initial value.

In Bentler’s notation, estimated coefficients are indicated by asterisks. Referring to a parameter in Bentler’s notation requires the specification of two variable names that correspond to the row and column of the position of the parameter in the matrix. Specifying the estimated coefficients by parameter names makes it easier to impose additional constraints with code. You do not need any additional statements to express equality constraints. Simply specify the same name for parameters that should have equal values.

If your model contains many unconstrained parameters and it is too cumbersome to find different parameter names, you can specify all those parameters by the same prefix name. A prefix is a short name followed by a colon. The CALIS procedure then generates a parameter name by appending an integer suffix to this prefix name. The prefix name should have no more than five or six characters so that the generated parameter name is not longer than eight characters. To avoid unintentional equality constraints, the prefix names should not coincide with explicitly defined parameter names.

For example, you can specify confirmatory second-order factor analysis model (mentioned on page 441)

\[ S = F_1 F_2 P_2 F_2^t F_1^t + F_1 U_2^2 F_1^t + U_1^2 \]

by using the LINEQS and STD statements:

```
LINEQS
V1 = X1 F1 + E1,
V2 = X2 F1 + E2,
V3 = X3 F1 + E3,
V4 = X4 F2 + E4,
V5 = X5 F2 + E5,
V6 = X6 F2 + E6,
V7 = X7 F3 + E7,
V8 = X8 F3 + E8,
V9 = X9 F3 + E9,
F1 = Y1 F4 + D1,
F2 = Y1 F4 + Y2 F5 + D2,
F3 = Y2 F5 + D3;
STD
E1-E9 = 9 * U:,
D1-D3 = 3 * V:,
F4 F5 = 2 * P;
RUN;
```
STD Statement

**STD assignment < , assignment... > ;**

where *assignment* represents *variables* = *pattern-definition*

The STD statement tells which variances are parameters to estimate and which are fixed. The STD statement can be used only with the LINEQS statement. You can specify only one STD statement with each LINEQS model statement. The STD statement defines the diagonal elements of the central model matrix $\Phi$. These elements correspond to the variances of the exogenous variables and to the error variances of the endogenous variables. Elements that are not defined are assumed to be 0.

Each *assignment* consists of a variable list (*variables*) on the left-hand side and a pattern list (*pattern-definition*) on the right-hand side of an equal sign. The *assignments* in the STD statement must be separated by commas. The *variables* list on the left-hand side of the equal sign should contain only names of variables that do not appear on the left-hand side of an equation in the LINEQS statement, that is, exogenous, error, and disturbance variables.

The *pattern-definition* on the right-hand side is similar to that used in the MATRIX statement. Each list element on the right-hand side defines the variance of the variable on the left-hand side in the same list position. A name on the right-hand side means that the corresponding variance is a parameter to estimate. A name on the right-hand side can be followed by a number inside parentheses that gives the initial value. A number on the right-hand side means that the corresponding variance of the variable on the left-hand side is fixed. If the right-hand-side list is longer than the left-hand-side variable list, the right-hand-side list is shortened to the length of the variable list. If the right-hand-side list is shorter than the variable list, the right-hand-side list is filled with repetitions of the last item in the list.

The right-hand side can also contain prefixes. A prefix is a short name followed by a colon. The CALIS procedure then generates a parameter name by appending an integer suffix to this prefix name. The prefix name should have no more than five or six characters so that the generated parameter name is not longer than eight characters. To avoid unintentional equality constraints, the prefix names should not coincide with explicitly defined parameter names. For example, if the prefix A is not used in any previous statement, this STD statement

**STD E1-E6=6 * A: (6 * 3.) ;**

defines the six error variances as free parameters $A1, \ldots, A6$, all with starting values of 3.
COV Statement

**COV** assignment < , assignment... > ;

where **assignment** represents variables < * variables2 > = pattern-definition

The COV statement tells which covariances are parameters to estimate and which are fixed. The COV statement can be used only with the LINEQS statement. The COV statement differs from the STD statement only in the meaning of the left-hand-side **variables** list. You can specify only one COV statement with each LINEQS statement. The COV statement defines the off-diagonal elements of the central model matrix $\Phi$. These elements correspond to the covariances of the exogenous variables and to the error covariances of the endogenous variables. Elements that are not defined are assumed to be 0. The **assignments** in the COV statement must be separated by commas.

The **variables** list on the left-hand side of the equal sign should contain only names of variables that do not appear on the left-hand side of an equation in the LINEQS statement, that is, exogenous, error, and disturbance variables.

The **pattern-definition** on the right-hand side is similar to that used in the MATRIX statement. Each list element on the right-hand side defines the covariance of a pair of variables in the list on the left-hand side. A name on the right-hand side can be followed by a number inside parentheses that gives the initial value. A number on the right-hand side means that the corresponding covariance of the variable on the left-hand side is fixed. If the right-hand-side list is longer than the left-hand-side variable list, the right-hand-side list is shortened to the length of the variable list. If the right-hand-side list is shorter than the variable list, the right-hand-side list is filled with repetitions of the last item in the list.

You can use one of two alternatives to refer to parts of $\Phi$. The first alternative uses only one variable list and refers to all distinct pairs of variables within the list. The second alternative uses two variable lists separated by an asterisk and refers to all pairs of variables among the two lists.

**Within-List Covariances**

Using $k$ variable names in the **variables** list on the left-hand side of an equal sign in a COV statement means that the parameter list (pattern-definition) on the right-hand side refers to all $k(k - 1)/2$ distinct variable pairs in the below-diagonal part of the $\Phi$ matrix. Order is very important. The order relation between the left-hand-side variable pairs and the right-hand-side parameter list is illustrated by the following example:

**COV** E1-E4 = PHI1-PHI6 ;

This corresponds to the following parameter setting:

- $[E2, E1] = PHI1,$
- $[E3, E1] = PHI2,$
- $[E3, E2] = PHI3,$
- $[E4, E1] = PHI4,$
- $[E4, E2] = PHI5,$
Chapter 17. The CALIS Procedure

The symmetric elements are generated automatically. When you use prefix names on the right-hand sides, you do not have to count the exact number of parameters. For example,

\[
\text{COV } E1-E4 = \text{PHI: ;}
\]

generates the same list of parameter names if the prefix PHI is not used in a previous statement.

\[
\begin{array}{c}
E1 \\
E2 \\
E3 \\
E4
\end{array}
\begin{array}{c}
\phi_1 \\
\phi_2 \phi_3 \\
\phi_4 \phi_5 \phi_6
\end{array}
\]

Within List Covariances

\[
\begin{array}{c}
E1 \\
E2 \\
E3 \\
E4
\end{array}
\begin{array}{c}
\phi_1 \phi_2 \\
\phi_3 \phi_4
\end{array}
\]

Between List Covariances

\[
\text{Figure 17.4. Within-List and Between-List Covariances}
\]

**Between-List Covariances**

Using \(k_1\) and \(k_2\) variable names in the two lists (separated by an asterisk) on the left-hand side of an equal sign in a COV statement means that the parameter list on the right-hand side refers to all \(k_1 \times k_2\) distinct variable pairs in the \(\Phi\) matrix. Order is very important. The order relation between the left-hand-side variable pairs and the right-hand-side parameter list is illustrated by the following example:

\[
\text{COV } E1 \ E2 \ E3\ \text{E4 = PHI1-PHI4 ;}
\]

This corresponds to the following parameter setting:

\[
[E1,E3]=\text{PHI1}, \ [E1,E4]=\text{PHI2}, \\
[E2,E3]=\text{PHI3}, \ [E2,E4]=\text{PHI4}.
\]

The symmetric elements are generated automatically.

Using prefix names on the right-hand sides prevents you from having to count the exact number of parameters.
FACTOR Model Statement

**FACTOR** < options > ;

You can use the FACTOR statement to specify an exploratory or confirmatory first-order factor analysis of the given covariance or correlation matrix \( C \),

\[
C = \mathbf{F}\mathbf{F}' + \mathbf{U}, \quad \mathbf{U} = \text{diag}
\]

or

\[
C = \mathbf{F}\mathbf{P}\mathbf{F}' + \mathbf{U}, \quad \mathbf{P} = \mathbf{P}'
\]

where \( \mathbf{U} \) is a diagonal matrix and \( \mathbf{P} \) is symmetric. Within this section, \( n \) denotes the number of manifest variables corresponding to the rows and columns of matrix \( C \), and \( m \) denotes the number of latent variables (factors or components) corresponding to the columns of the loading matrix \( \mathbf{F} \).

You can specify only one FACTOR statement with each PROC CALIS statement. You can specify higher-order factor analysis problems using a COSAN model specification. PROC CALIS requires more computing time and memory than PROC FACTOR because it is designed for more general structural estimation problems and is unable to exploit the special properties of the unconstrained factor analysis model.

For default (exploratory) factor analysis, PROC CALIS computes initial estimates for factor loadings and unique variances by an algebraic method of approximate factor analysis. If you use a MATRIX statement together with a FACTOR model specification, initial values are computed by McDonald’s (McDonald and Hartmann 1992) method (if possible). For details, see the section “Using the FACTOR and MATRIX Statements” on page 442. If neither of the two methods are appropriate, the initial values are set by the START= option.

The unrestricted factor analysis model is not identified because any orthogonal rotated factor loading matrix \( \tilde{\mathbf{F}} = \mathbf{F}\mathbf{\Theta} \) is equivalent to the result \( \mathbf{F} \),

\[
C = \tilde{\mathbf{F}}\tilde{\mathbf{F}}' + \mathbf{U}, \quad \tilde{\mathbf{F}} = \mathbf{F}\mathbf{\Theta}, \quad \text{where} \quad \mathbf{\Theta}'\mathbf{\Theta} = \mathbf{\Theta}\mathbf{\Theta}' = \mathbf{I}
\]

To obtain an identified factor solution, the FACTOR statement imposes zero constraints on the \( m(m - 1)/2 \) elements in the upper triangle of \( \mathbf{F} \) by default.

The following options are available in the FACTOR statement.

**COMPONENT | COMPSTART**

computes a component analysis instead of a factor analysis (the diagonal matrix \( \mathbf{U} \) in the model is set to 0). Note that the rank of \( \mathbf{F}\mathbf{F}' \) is equal to the number \( m \) of components in \( \mathbf{F} \). If \( m \) is smaller than the number of variables in the moment matrix \( C \), the matrix of predicted model values is singular and maximum likelihood estimates for \( \mathbf{F} \) cannot be computed. You should compute ULS estimates in this case.
CONSTRUCTION

Heywood | Hey

constrains the diagonal elements of $U$ to be nonnegative; in other words, the model is replaced by

$$C = FF' + U^2, \quad U = \text{diag}$$

$N = m$

specifies the number of first-order factors or components. The number $m$ of factors should not exceed the number $n$ of variables in the covariance or correlation matrix analyzed. For the saturated model, $m = n$, the COMP option should generally be specified for $U = 0$; otherwise, $df < 0$. For $m = 0$ no factor loadings are estimated, and the model is $C = U$, with $U = \text{diag}$. By default, $m = 1$.

NORM

normalizes the rows of the factor pattern for rotation using Kaiser’s normalization.

ROBUST | R = name

specifies an orthogonal rotation. By default, ROBUST=NONE. The possible values for name are as follows:

PRINCIPAL | PC specifies a principal axis rotation. If ROBUST=PRINCIPAL is used with a factor rather than a component model, the following rotation is performed:

$$F_{\text{new}} = F_{\text{old}}T, \quad \text{with} \quad F'_{\text{old}}F_{\text{old}} = \Lambda T'$$

where the columns of matrix $T$ contain the eigenvectors of $F'_{\text{old}}F_{\text{old}}$.

QUARTIMAX | Q specifies quartimax rotation.

VARIMAX | V specifies varimax rotation.

EQUAMAX | E specifies equamax rotation.

PARSIMAX | P specifies parsimax rotation.

NONE performs no rotation (default).

**BOUNDS Statement**

**BOUNDS constraint < , constraint ... > ;**

where constraint represents

$$< \text{number operator} > \text{parameter-list} < \text{operator number} >$$

You can use the BOUNDS statement to define boundary constraints for any parameter that has its name specified in a MATRIX, LINEQS, STD, COV, or RAM statement or that is used in the model of an INRAM= data set. Valid operators are $<=$, $<$, $>$=, $>$, and $=$ or, equivalently, LE, LT, GE, GT, and EQ. The following is an example of the BOUNDS statement:
bounds 0. <= a1-a9 x  <= 1. ,
  -1. <= c2-c5 ,
  b1-b10 y >= 0. ;

You must separate boundary constraints with a comma, and you can specify more than one BOUNDS statement. The feasible region for a parameter is the intersection of all boundary constraints specified for that parameter; if a parameter has a maximum lower boundary constraint larger than its minimum upper bound, the parameter is set equal to the minimum of the upper bounds.

If you need to compute the values of the upper or lower bounds, create a TYPE=EST data set containing _TYPE_='UPPERBD' or _TYPE_='LOWERBD' observations and use it as an INEST= or INVAR= input data set in a later PROC CALIS run.

The BOUNDS statement can contain only parameter names and numerical constants. You cannot use the names of variables created in program statements.

The active set strategies made available in PROC CALIS cannot realize the strict inequality constraints < or >. For example, you cannot specify BOUNDS x > 0; to prevent infinite values for y = log(x). Use BOUNDS x > 1E-8; instead.

If the CALIS procedure encounters negative diagonal elements in the central model matrices during the minimization process, serious convergence problems can occur. You can use the BOUNDS statement to constrain these parameters to nonnegative values. Using negative values in these locations can lead to a smaller \( \chi^2 \) value but uninterpretable estimates.

**LINCON Statement**

**LINCON** constraint < , constraint ... > ;
where constraint represents
number operator linear-term or
linear-term operator number ,
and linear-term is

\(<|>|<\text{coefficient}^*|>|\text{parameter}<|>|<\text{coefficient}^*|>|\text{parameter}...>\)

The LINCON statement specifies a set of linear equality or inequality constraints of the form

\[
\sum_{j=1}^{n} a_{ij} x_j \leq b_i, \quad i = 1, \ldots, m
\]

The constraints must be separated by commas. Each linear constraint \( i \) in the statement consists of a linear combination \( \sum_j a_{ij} x_j \) of a subset of the \( n \) parameters \( x_j, j = 1, \ldots, n \), and a constant value \( b_i \) separated by a comparison operator. Valid operators are \(<=, <, >, =, >\), and = or, equivalently, LE, LT, GE, GT, and EQ. PROC CALIS cannot enforce the strict inequalities < or >. Note that the coefficients \( a_{ij} \) in the linear combination must be constant numbers and must be followed by an asterisk and the name of a parameter (listed in the PARMS or VAR statement).
Although you can easily express boundary constraints in LINCON statements, for many applications it is much more convenient to specify both the BOUNDS and the LINCON statements in the same PROC CALIS call.

The LINCON statement can contain only parameter names, operators, and numerical constants. If you need to compute the values of the coefficients $a_{ij}$ or right-hand sides $b_i$, you can run a preliminary DATA step and create a TYPE=EST data set containing _TYPE_='LE', _TYPE_='GE', or _TYPE_='EQ' observations, then specify this data set as an INEST= or INVAR= data set in a following PROC CALIS run.

### NLINCON Statement

**NLINCON | NLC constraint < , constraint ... > ;**

where constraint represents

- number operator variable-list number operator or
- variable-list operator number or
- number operator variable-list

You can specify nonlinear equality and inequality constraints with the NLINCON or NLC statement. The QUANEW optimization subroutine enables you to specify nonlinear constraints using the NLINCON statement.

The syntax of the NLINCON statement is similar to that of the BOUNDS statement, except that the NLINCON statement can contain the names of variables that are continuous functions of the parameters. The values of the nonlinear constraint variables must be computed in the program statements.

As with the BOUNDS statement, one- or two-sided constraints are allowed in the NLINCON statement; equality constraints must be one sided. Valid operators are $<=$, $<$, $>$, $$ at, and $==$ or, equivalently, LE, LT, GE, GT, and EQ.

PROC CALIS cannot enforce the strict inequalities $<$ or $>$ but instead treats them as $<=$ and $$ at, respectively. The listed nonlinear constraints must be separated by commas.

The following three NLINCON statements, which require the values of the three constraint variables V1, V2, and V3 to be between zero and ten, are equivalent:

```
NLINCON 0. <= V1-V3,
        V1-V3 <= 10;
NLINCON 0. <= V1-V3 <= 10.;
NLINCON 10. >= V1-V3 >= 0.;
```
NLOPTIONS Statement

NLOPTIONS option(s) ;

Many options that are available in PROC NLP can now be specified for the optimization subroutines in PROC CALIS using the NLOPTIONS statement. The NLOPTIONS statement provides more displayed and file output on the results of the optimization process, and it permits the same set of termination criteria as in PROC NLP. The available options are summarized in Table 17.2 through Table 17.4, and the options are described in detail in the following three sections.

Table 17.2. Options Documented in the PROC CALIS Statement

<table>
<thead>
<tr>
<th>Option</th>
<th>Short Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Estimation Methods</strong></td>
<td></td>
</tr>
<tr>
<td>G4=i</td>
<td>algorithm for computing STDERR</td>
</tr>
<tr>
<td><strong>Optimization Techniques</strong></td>
<td></td>
</tr>
<tr>
<td>TECHNIQUE=name</td>
<td>minimization method</td>
</tr>
<tr>
<td>UPDATE=name</td>
<td>update technique</td>
</tr>
<tr>
<td>LINESEARCH=i</td>
<td>line-search method</td>
</tr>
<tr>
<td>FCONV=r</td>
<td>relative change function convergence criterion</td>
</tr>
<tr>
<td>GCONV=r</td>
<td>relative gradient convergence criterion</td>
</tr>
<tr>
<td>INSTEP=r</td>
<td>initial step length (SALPHA=, RADIUS=)</td>
</tr>
<tr>
<td>LSPRECISION=r</td>
<td>line-search precision</td>
</tr>
<tr>
<td>MAXFUNC=i &lt;n&gt;</td>
<td>maximum number of function calls</td>
</tr>
<tr>
<td>MAXITER=i &lt;n&gt;</td>
<td>maximum number of iterations</td>
</tr>
<tr>
<td><strong>Miscellaneous Options</strong></td>
<td></td>
</tr>
<tr>
<td>ASINGULAR=r</td>
<td>absolute singularity criterion for inversion of the information matrix</td>
</tr>
<tr>
<td>COVSING=r</td>
<td>singularity tolerance of the information matrix</td>
</tr>
<tr>
<td>MSINGULAR=r</td>
<td>relative M singularity criterion for inversion of the information matrix</td>
</tr>
<tr>
<td>SINGULAR=r</td>
<td>singularity criterion for inversion of the Hessian</td>
</tr>
<tr>
<td>VSINGULAR=r</td>
<td>relative V singularity criterion for inversion of the information matrix</td>
</tr>
</tbody>
</table>
### Table 17.3. Termination Criteria Options

<table>
<thead>
<tr>
<th>Option</th>
<th>Short Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Options Used by All Techniques</strong></td>
<td></td>
</tr>
<tr>
<td>ABSCONV=r</td>
<td>absolute function convergence criterion</td>
</tr>
<tr>
<td>MAXFUNC=i</td>
<td>maximum number of function calls</td>
</tr>
<tr>
<td>MAXITER=i &lt;n&gt;</td>
<td>maximum number of iterations</td>
</tr>
<tr>
<td>MAXTIME=r</td>
<td>maximum CPU time</td>
</tr>
<tr>
<td>MINITER=i</td>
<td>minimum number of iterations</td>
</tr>
<tr>
<td><strong>Options for Unconstrained and Linearly Constrained Techniques</strong></td>
<td></td>
</tr>
<tr>
<td>ABSFCONV=r &lt;n&gt;</td>
<td>absolute change function convergence criterion</td>
</tr>
<tr>
<td>ABSGCONV=r &lt;n&gt;</td>
<td>absolute gradient convergence criterion</td>
</tr>
<tr>
<td>ABSXCONV=r &lt;n&gt;</td>
<td>absolute change parameter convergence criterion</td>
</tr>
<tr>
<td>FCONV=r &lt;n&gt;</td>
<td>relative change function convergence criterion</td>
</tr>
<tr>
<td>FCONV2=r &lt;n&gt;</td>
<td>function convergence criterion</td>
</tr>
<tr>
<td>FDIGITS=r</td>
<td>precision in computation of the objective function</td>
</tr>
<tr>
<td>FSIZE=r</td>
<td>parameter for FCONV= and GCONV=</td>
</tr>
<tr>
<td>GCONV=r &lt;n&gt;</td>
<td>relative gradient convergence criterion</td>
</tr>
<tr>
<td>GCONV2=r &lt;n&gt;</td>
<td>relative gradient convergence criterion</td>
</tr>
<tr>
<td>XCONV=r &lt;n&gt;</td>
<td>relative change parameter convergence criterion</td>
</tr>
<tr>
<td>XSIZE=r</td>
<td>parameter for XCONV=</td>
</tr>
<tr>
<td><strong>Options for Nonlinearly Constrained Techniques</strong></td>
<td></td>
</tr>
<tr>
<td>ABSGCONV=r &lt;n&gt;</td>
<td>maximum absolute gradient of Lagrange function criterion</td>
</tr>
<tr>
<td>FCONV2=r &lt;n&gt;</td>
<td>predicted objective function reduction criterion</td>
</tr>
<tr>
<td>GCONV=r &lt;n&gt;</td>
<td>normalized predicted objective function reduction criterion</td>
</tr>
</tbody>
</table>
Table 17.4. Miscellaneous Options

<table>
<thead>
<tr>
<th>Option</th>
<th>Short Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Options for the Approximate Covariance Matrix of Parameter Estimates</strong></td>
<td></td>
</tr>
<tr>
<td>CFACTOR=r</td>
<td>scalar factor for STDERR</td>
</tr>
<tr>
<td>NOHLF</td>
<td>use Hessian of the objective function for STDERR</td>
</tr>
<tr>
<td><strong>Options for Additional Displayed Output</strong></td>
<td></td>
</tr>
<tr>
<td>PALL</td>
<td>display initial and final optimization values</td>
</tr>
<tr>
<td>PCRPJAC</td>
<td>display approximate Hessian matrix</td>
</tr>
<tr>
<td>PHESSIAN</td>
<td>display Hessian matrix</td>
</tr>
<tr>
<td>PHISTORY</td>
<td>display optimization history</td>
</tr>
<tr>
<td>PINIT</td>
<td>display initial values and derivatives (PALL)</td>
</tr>
<tr>
<td>PNLCJAC</td>
<td>display Jacobian matrix of nonlinear constraints (PALL)</td>
</tr>
<tr>
<td>PRINT</td>
<td>display results of the optimization process</td>
</tr>
<tr>
<td><strong>Additional Options for Optimization Techniques</strong></td>
<td></td>
</tr>
<tr>
<td>DAMPSTEP&lt; =r &gt;</td>
<td>controls initial line-search step size</td>
</tr>
<tr>
<td>HESCAL=n</td>
<td>scaling version of Hessian or Jacobian</td>
</tr>
<tr>
<td>LCDEACT=r</td>
<td>Lagrange multiplier threshold of constraint</td>
</tr>
<tr>
<td>LCEPSILON=r</td>
<td>range for boundary and linear constraints</td>
</tr>
<tr>
<td>LCSINGULAR=r</td>
<td>QR decomposition linear dependence criterion</td>
</tr>
<tr>
<td>NOEIGNUM</td>
<td>suppress computation of matrices</td>
</tr>
<tr>
<td>RESTART=i</td>
<td>restart algorithm with a steepest descent direction</td>
</tr>
<tr>
<td>VERSION=1</td>
<td>quasi-Newton optimization technique version</td>
</tr>
</tbody>
</table>

Options Documented in the PROC CALIS Statement

The following options are the same as in the PROC CALIS statement and are documented in the section “PROC CALIS Statement” on page 415.

Estimation Method Option

G4=i

specifies the method for computing the generalized (G2 or G4) inverse of a singular matrix needed for the approximate covariance matrix of parameter estimates. This option is valid only for applications where the approximate covariance matrix of parameter estimates is found to be singular.
**Optimization Technique Options**

TECHNIQUE | TECH=name

OMETHOD | OM=name

specifies the optimization technique.

UPDATE | UPD=name

specifies the update method for the quasi-Newton or conjugate-gradient optimization technique.

LINESEARCH | LIS=i

specifies the line-search method for the CONGRA, QUANEW, and NEWRAP optimization techniques.

FCONV | FTOL=r

specifies the relative function convergence criterion. For more details, see the section “Termination Criteria Options” on page 461.

GCONV | GTOL=r

specifies the relative gradient convergence criterion. For more details, see the section “Termination Criteria Options” on page 461.

INSTEP | SALPHA | RADIUS=r

restricts the step length of an optimization algorithm during the first iterations.

LSPRECISION | LSP=r

specifies the degree of accuracy that should be obtained by the line-search algorithms LIS=2 and LIS=3.

MAXFUNC | MAXFU=i

specifies the maximum number i of function calls in the optimization process. For more details, see the section “Termination Criteria Options” on page 461.

MAXITER | MAXIT=i <n>

specifies the maximum number i of iterations in the optimization process. For more details, see the section “Termination Criteria Options” on page 461.

**Miscellaneous Options**

ASINGULAR | ASING=r

specifies an absolute singularity criterion r, r > 0, for the inversion of the information matrix, which is needed to compute the approximate covariance matrix of parameter estimates.

COVSING=r

specifies a nonnegative threshold r, r > 0, that decides whether the eigenvalues of the information matrix are considered to be zero. This option is valid only for applications where the approximate covariance matrix of parameter estimates is found to be singular.

MSINGULAR | MSING=r

specifies a relative singularity criterion r, r > 0, for the inversion of the information matrix, which is needed to compute the approximate covariance matrix of parameter estimates.
SINGULAR | SING = r
specifies the singularity criterion \( r \), \( 0 \leq r \leq 1 \), that is used for the inversion of the Hessian matrix. The default value is 1E−8.

VSINGULAR | VSING = r
specifies a relative singularity criterion \( r \), \( r > 0 \), for the inversion of the information matrix, which is needed to compute the approximate covariance matrix of parameter estimates.

**Termination Criteria Options**

Let \( x^* \) be the point at which the objective function \( f(.) \) is optimized, and let \( x^{(k)} \) be the parameter values attained at the \( k \)th iteration. All optimization techniques stop at the \( k \)th iteration if at least one of a set of termination criteria is satisfied. The specified termination criteria should allow termination in an area of sufficient size around \( x^* \). You can avoid termination respective to any of the following function, gradient, or parameter criteria by setting the corresponding option to zero. There is a default set of termination criteria for each optimization technique; most of these default settings make the criteria ineffective for termination. PROC CALIS may have problems due to rounding errors (especially in derivative evaluations) that prevent an optimizer from satisfying strong termination criteria.

Note that PROC CALIS also terminates if the point \( x^{(k)} \) is fully constrained by linearly independent active linear or boundary constraints, and all Lagrange multiplier estimates of active inequality constraints are greater than a small negative tolerance.

The following options are available only in the NLOPTIONS statement (except for FCONV, GCONV, MAXFUNC, and MAXITER), and they affect the termination criteria.

**Options Used by All Techniques**
The following five criteria are used by all optimization techniques.

**ABSCONV | ABSTOL = r**
specifies an absolute function convergence criterion.
- For minimization, termination requires \( f^{(k)} = f(x^{(k)}) \leq ABSCONV \)
- For maximization, termination requires \( f^{(k)} = f(x^{(k)}) \geq ABSCONV \)

The default value of ABSCONV is
- for minimization, the negative square root of the largest double precision value
- for maximization, the positive square root of the largest double precision value

**MAXFUNC | MAXFU = i**
requires the number of function calls to be no larger than \( i \). The default values are listed in the following table.

<table>
<thead>
<tr>
<th>TECH=</th>
<th>MAXFUNC default</th>
</tr>
</thead>
<tbody>
<tr>
<td>LEVMAR, NEWRAP, NRRIDG, TRUREG</td>
<td>( i=125 )</td>
</tr>
<tr>
<td>DBLDOG, QUANEW</td>
<td>( i=500 )</td>
</tr>
<tr>
<td>CONGRA</td>
<td>( i=1000 )</td>
</tr>
</tbody>
</table>
The default is used if you specify MAXFUNC=0. The optimization can be terminated only after completing a full iteration. Therefore, the number of function calls that is actually performed can exceed the number that is specified by the MAXFUNC= option.

**MAXITER | MAXIT= i <n>**

requires the number of iterations to be no larger than \( i \). The default values are listed in the following table.

<table>
<thead>
<tr>
<th>TECH=</th>
<th>MAXITER default</th>
</tr>
</thead>
<tbody>
<tr>
<td>LEVMAR, NEWRAP, NRRIDG, TRUREG</td>
<td>( i=50 )</td>
</tr>
<tr>
<td>DBLDOG, QUANEW</td>
<td>( i=200 )</td>
</tr>
<tr>
<td>CONGRA</td>
<td>( i=400 )</td>
</tr>
</tbody>
</table>

The default is used if you specify MAXITER=0 or missing.

The optional second value \( n \) is valid only for TECH=QUANEW with nonlinear constraints. It specifies an upper bound \( n \) for the number of iterations of an algorithm and reduces the violation of nonlinear constraints at a starting point. The default value is \( n=20 \). For example, specifying MAXITER= . 0 means that you do not want to exceed the default number of iterations during the main optimization process and that you want to suppress the feasible point algorithm for nonlinear constraints.

**MAXTIME= r**

requires the CPU time to be no larger than \( r \). The default value of the MAXTIME= option is the largest double floating point number on your computer.

**MINITER | MINIT= i**

specifies the minimum number of iterations. The default value is \( i = 0 \).

The ABSCONV=, MAXITER=, MAXFUNC=, and MAXTIME= options are useful for dividing a time-consuming optimization problem into a series of smaller problems by using the OUTEST= and INEST= data sets.

### Options for Unconstrained and Linearly Constrained Techniques

This section contains additional termination criteria for all unconstrained, boundary, or linearly constrained optimization techniques.

**ABSFCONV | ABSFTOL= r <n>**

specifies the absolute function convergence criterion. Termination requires a small change of the function value in successive iterations,

\[
| f(x^{(k-1)}) - f(x^{(k)}) | \leq r
\]

The default value is \( r = 0 \). The optional integer value \( n \) determines the number of successive iterations for which the criterion must be satisfied before the process can be terminated.

**ABSGCONV | ABSGTOL= r <n>**

specifies the absolute gradient convergence criterion. Termination requires the maximum absolute gradient element to be small,

\[
\max_j | g_j^{(k)} | \leq r
\]
The default value is \( r = 1E-5 \). The optional integer value \( n \) determines the number of successive iterations for which the criterion must be satisfied before the process can be terminated.

Note: In some applications, the small default value of the ABSGCONV= criterion is too difficult to satisfy for some of the optimization techniques.

**ABSXCONV | ABSXTOL**

\( r < n \)

specifies the absolute parameter convergence criterion. Termination requires a small Euclidean distance between successive parameter vectors,

\[
\| x^{(k)} - x^{(k-1)} \|_2 \leq r
\]

The default value is \( r = 0 \). The optional integer value \( n \) determines the number of successive iterations for which the criterion must be satisfied before the process can be terminated.

**FCONV | FTOL**

\( r < n \)

specifies the relative function convergence criterion. Termination requires a small relative change of the function value in successive iterations,

\[
\frac{|f(x^{(k)}) - f(x^{(k-1)})|}{\max(|f(x^{(k-1)})|, FSIZE)} \leq r
\]

where \( FSIZE \) is defined by the FSIZE= option. The default value is \( r = 10^{-FDIGITS} \), where \( FDIGITS \) either is specified or is set by default to \(-\log_{10}(\epsilon)\), where \( \epsilon \) is the machine precision. The optional integer value \( n \) determines the number of successive iterations for which the criterion must be satisfied before the process can be terminated.

**FCONV2 | FTOL2**

\( r < n \)

specifies another function convergence criterion. For least-squares problems, termination requires a small predicted reduction

\[
df^{(k)} \approx f(x^{(k)}) - f(x^{(k)} + s^{(k)})
\]

of the objective function.

The predicted reduction

\[
df^{(k)} = -g^{(k)'s^{(k)}} - \frac{1}{2}s^{(k)'G^{(k)}s^{(k)}} = -\frac{1}{2}s^{(k)'g^{(k)}} \leq r
\]
is computed by approximating the objective function $f$ by the first two terms of the Taylor series and substituting the Newton step

$$s^{(k)} = -G^{(k)}^{-1}g^{(k)}$$

The FCONV2 criterion is the unscaled version of the GCONV criterion. The default value is $r = 0$. The optional integer value $n$ determines the number of successive iterations for which the criterion must be satisfied before the process can be terminated.

**FDIGITS=$r$**

specifies the number of accurate digits in evaluations of the objective function. Fractional values such as FDIGITS=4.7 are allowed. The default value is $r = -\log_{10}\epsilon$, where $\epsilon$ is the machine precision. The value of $r$ is used for the specification of the default value of the FCONV= option.

**FSIZE=$r$**

specifies the FSIZE parameter of the relative function and relative gradient termination criteria. The default value is $r = 0$. See the FCONV= and GCONV= options.

**GCONV | GTOL=$r<n>$**

specifies the relative gradient convergence criterion. For all techniques except the CONGRA technique, termination requires that the normalized predicted function reduction is small,

$$\frac{[g^{(k)}]'^T[G^{(k)}]^{-1}g^{(k)}}{\max([f(x^{(k)})], FSIZE)} \leq r$$

where FSIZE is defined by the FSIZE= option. For the CONGRA technique (where a reliable Hessian estimate $G$ is not available),

$$\frac{\| g^{(k)} \|_2^2 - \| s^{(k)} \|_2}{\max([f(x^{(k)})], FSIZE)} \leq r$$

is used. The default value is $r = 1E-8$. The optional integer value $n$ determines the number of successive iterations for which the criterion must be satisfied before the process can be terminated.

Note: The default setting for the GCONV= option sometimes leads to early termination far from the location of the optimum. This is especially true for the special form of this criterion used in the CONGRA optimization.

**GCONV2 | GTOL2=$r<n>$**

specifies another relative gradient convergence criterion. For least-squares problems and the TRUREG, LEVMAR, NRRIDG, and NEWRAP techniques, the criterion of Browne (1982) is used,

$$\max_j \frac{|g_j^{(k)}|}{\sqrt{f(x^{(k)})G_{j,j}^{(k)}}} \leq r$$
This criterion is not used by the other techniques. The default value is $r = 0$. The optional integer value $n$ determines the number of successive iterations for which the criterion must be satisfied before the process can be terminated.

\[ XCONV | XTOL=r < n > \]

specifies the relative parameter convergence criterion. Termination requires a small relative parameter change in subsequent iterations,

\[
\frac{\max_j |x_j^{[k]} - x_j^{[k-1]}|}{\max(|x_j^{[k]}|, |x_j^{[k-1]}|, XSIZE)} \leq r
\]

The default value is $r = 0$. The optional integer value $n$ determines the number of successive iterations for which the criterion must be satisfied before the process can be terminated.

\[ XSIZE=r \]

specifies the $XSIZE$ parameter of the relative function and relative gradient termination criteria. The default value is $r = 0$. See the XCONV= option.

**Options for Nonlinearly Constrained Techniques**

The non-NMSIMP algorithms available for nonlinearly constrained optimization (currently only TECH=QUANEW) do not monotonically reduce either the value of the objective function or some kind of merit function that combines objective and constraint functions. Furthermore, the algorithm uses the watchdog technique with backtracking (Chamberlain et al., 1982). Therefore, no termination criteria are implemented that are based on the values ($x$ or $f$) of successive iterations. In addition to the criteria used by all optimization techniques, only three more termination criteria are currently available, and they are based on the Lagrange function

\[ L(x, \lambda) = f(x) - \sum_{i=1}^{m} \lambda_i c_i(x) \]

and its gradient

\[ \nabla_x L(x, \lambda) = g(x) - \sum_{i=1}^{m} \lambda_i \nabla_x c_i(x) \]

Here, $m$ denotes the total number of constraints, $g = g(x)$ denotes the gradient of the objective function, and $\lambda$ denotes the $m$ vector of Lagrange multipliers. The Kuhn-Tucker conditions require that the gradient of the Lagrange function is zero at the optimal point $(x^*, \lambda^*)$:

\[ \nabla_x L(x^*, \lambda^*) = 0 \]

The termination criteria available for nonlinearly constrained optimization follow.
Chapter 17. The CALIS Procedure

**ABSGCONV | ABSGTOL** = \( r < n > 

specifies that termination requires the maximum absolute gradient element of the Lagrange function to be small,

\[
\max_j |\{\nabla_x L(x^{(k)}, \lambda^{(k)})\}_j| \leq r
\]

The default value is \( r=1E-5 \). The optional integer value \( n \) determines the number of successive iterations for which the criterion must be satisfied before the process can be terminated.

**FCONV2 | FTOL2** = \( r < n > 

specifies that termination requires the predicted objective function reduction to be small:

\[
|g(x^{(k)})s(x^{(k)})| + \sum_{i=1}^{m} |\lambda_i c_i| \leq r
\]

The default value is \( r=1E-6 \). This is the criterion used by the programs VMCWD and VF02AD (Powell 1982b). The optional integer value \( n \) determines the number of successive iterations for which the criterion must be satisfied before the process can be terminated.

**GCONV | GTOL** = \( r < n > 

specifies that termination requires the normalized predicted objective function reduction to be small:

\[
\frac{|g(x^{(k)})s(x^{(k)})| + \sum_{i=1}^{m} |\lambda_i c_i(x^{(k)})|}{\max(|f(x^{(k)})|,FSIZE)} \leq r
\]

where \( FSIZE \) is defined by the FSIZE= option. The default value is \( r=1E-8 \). The optional integer value \( n \) determines the number of successive iterations for which the criterion must be satisfied before the process can be terminated.

**Miscellaneous Options**

**Options for the Approximate Covariance Matrix of Parameter Estimates**

You can specify the following options to modify the approximate covariance matrix of parameter estimates.

**CFACOR=** \( r 

specifies the scalar factor for the covariance matrix of parameter estimates. The scalar \( r \geq 0 \) replaces the default value \( c/NM \). For more details, see the section “Approximate Standard Errors” on page 494.

**NOHLF**

specifies that the Hessian matrix of the objective function (rather than the Hessian matrix of the Lagrange function) is used for computing the approximate covariance matrix of parameter estimates and, therefore, the approximate standard errors.

It is theoretically not correct to use the NOHLF option. However, since most implementations use the Hessian matrix of the objective function and not the Hessian ma-
trix of the Lagrange function for computing approximate standard errors, the NOHLF option can be used to compare the results.

**Options for Additional Displayed Output**

You can specify the following options to obtain additional displayed output.

**PALL | ALL**

- displays information on the starting values and final values of the optimization process.

**PCRPJAC | PJTJ**

- displays the approximate Hessian matrix. If general linear or nonlinear constraints are active at the solution, the projected approximate Hessian matrix is also displayed.

**PHESSIAN | PHES**

- displays the Hessian matrix. If general linear or nonlinear constraints are active at the solution, the projected Hessian matrix is also displayed.

**PHISTORY | PHIS**

- displays the optimization history. The PHISTORY option is set automatically if the PALL or PRINT option is set.

**PINIT | PIN**

- displays the initial values and derivatives (if available). The PINIT option is set automatically if the PALL option is set.

**PNLCJAC**

- displays the Jacobian matrix of nonlinear constraints specified by the NLINCON statement. The PNLCJAC option is set automatically if the PALL option is set.

**PRINT | PRI**

- displays the results of the optimization process, such as parameter estimates and constraints.

**More Options for Optimization Techniques**

You can specify the following options, in addition to the options already listed, to fine-tune the optimization process. These options should not be necessary in most applications of PROC CALIS.

**DAMPSTEP | DS <\(=r\)>**

- specifies that the initial step-size value \(\alpha^{(0)}\) for each line search (used by the QUANEW, CONGRA, or NEWRAP techniques) cannot be larger than \(r\) times the step-size value used in the former iteration. If the factor \(r\) is not specified, the default value is \(r = 2\). The DAMPSTEP option can prevent the line-search algorithm from repeatedly stepping into regions where some objective functions are difficult to compute or where they can lead to floating point overflows during the computation of objective functions and their derivatives. The DAMPSTEP\(<=r\>\) option can prevent time-costly function calls during line searches with very small step sizes \(\alpha\) of objective functions. For more information on setting the start values of each line search, see the section “Restricting the Step Length” on page 517.
HESCAL | HS = 0 | 1 | 2 | 3
specifies the scaling version of the Hessian or crossproduct Jacobian matrix used in
NRRIDG, TRUREG, LEVMAR, NEWRAP, or DBLDOG optimization. If HS is not
equal to zero, the first iteration and each restart iteration sets the diagonal scaling
matrix \( D^{(0)} = \text{diag}(d_i^{(0)}) \):

\[
d_i^{(0)} = \sqrt{\max(|G_{i,i}^{(0)}|, \epsilon)}
\]

where \( G_{i,i}^{(0)} \) are the diagonal elements of the Hessian or crossproduct Jacobian matrix.

In every other iteration, the diagonal scaling matrix \( D^{(k)} = \text{diag}(d_i^{(k)}) \) is updated
depending on the HS option:

HS=0 specifies that no scaling is done.
HS=1 specifies the Moré (1978) scaling update:

\[
d_i^{(k+1)} = \max(d_i^{(k)}, \sqrt{\max(|G_{i,i}^{(k)}|, \epsilon)})
\]

HS=2 specifies the Dennis, Gay, and Welsch (1981) scaling update:

\[
d_i^{(k+1)} = \max(0.6 \cdot d_i^{(k)}, \sqrt{\max(|G_{i,i}^{(k)}|, \epsilon)})
\]

HS=3 specifies that \( d_i \) is reset in each iteration:

\[
d_i^{(k+1)} = \sqrt{\max(|G_{i,i}^{(k)}|, \epsilon)}
\]

In the preceding equations, \( \epsilon \) is the relative machine precision. The default is
HS=1 for LEVMAR minimization and HS=0 otherwise. Scaling of the Hessian or
crossproduct Jacobian can be time-consuming in the case where general linear con-
straints are active.

LCDEACT | LCD = r
specifies a threshold \( r \) for the Lagrange multiplier that decides whether an active
inequality constraint remains active or can be deactivated. For maximization, \( r \) must
be greater than zero; for minimization, \( r \) must be smaller than zero. The default is
\[
r = \pm \min(0.01, \max(0.1 \cdot ABSGCONV, 0.001 \cdot g_{max}^{(k)}))
\]

where “+” stands for maximization, “−” stands for minimization, \( ABSGCONV \) is
the value of the absolute gradient criterion, and \( g_{max}^{(k)} \) is the maximum absolute
element of the (projected) gradient \( g^{(k)} \) or \( Z^t g^{(k)} \).

LCEPSILON | LCEPS | LCE = r
specifies the range \( r, r \geq 0, \) for active and violated boundary and linear constraints.
If the point \( x^{(k)} \) satisfies the condition

\[
\left| \sum_{j=1}^{n} a_{ij} x_j^{(k)} - b_i \right| \leq r \cdot (|b_i| + 1)
\]
the constraint \( i \) is recognized as an active constraint. Otherwise, the constraint \( i \) is either an inactive inequality or a violated inequality or equality constraint. The default value is \( r=1E-8 \). During the optimization process, the introduction of rounding errors can force PROC NLP to increase the value of \( r \) by factors of 10. If this happens, it is indicated by a message displayed in the log.

**LCSINGULAR | LCSING | LCS = r**
specifies a criterion \( r, r \geq 0 \), used in the update of the QR decomposition that decides whether an active constraint is linearly dependent on a set of other active constraints. The default is \( r=1E-8 \). The larger \( r \) becomes, the more the active constraints are recognized as being linearly dependent.

**NOEIGNUM**
suppresses the computation and displayed output of the determinant and the inertia of the Hessian, crossproduct Jacobian, and covariance matrices. The inertia of a symmetric matrix are the numbers of negative, positive, and zero eigenvalues. For large applications, the NOEIGNUM option can save computer time.

**RESTART | REST = i**
specifies that the QUANEW or CONGRA algorithm is restarted with a steepest descent/ascent search direction after at most \( i \) iterations, \( i > 0 \). Default values are as follows:

- **CONGRA**: UPDATE=PB: restart is done automatically so specification of \( i \) is not used.
- **CONGRA**: UPDATE\( \neq \)PB: \( i = \min(10n, 80) \), where \( n \) is the number of parameters.
- **QUANEW**: \( i \) is the largest integer available.

**VERSION | VS = 1 | 2**
specifies the version of the quasi-Newton optimization technique with nonlinear constraints.

- **VS=1** specifies the update of the \( \mu \) vector as in Powell (1978a, 1978b) (update like VF02AD).
- **VS=2** specifies the update of the \( \mu \) vector as in Powell (1982a, 1982b) (update like VMCWD).

The default is VS=2.
PARAMETERS Statement

PARAMETERS | PARMS parameter(s) << = > number(s) >>
            << , > parameter(s) << = > number(s) > . . . >> ;

The PARAMETERS statement defines additional parameters that are not elements of a model matrix to use in your own program statements. You can specify more than one PARAMETERS statement with each PROC CALIS statement. The parameters can be followed by an equal sign and a number list. The values of the numbers list are assigned as initial values to the preceding parameters in the parameters list. For example, each of the following statements assigns the initial values ALPHA=.5 and BETA=-.5 for the parameters used in program statements:

PARAMETERS ALFA BETA=.5 -.5;
PARAMETERS ALFA BETA (.5 -.5);
PARAMETERS ALFA BETA .5 -.5;
PARAMETERS ALFA=.5 BETA (-.5);

The number of parameters and the number of values does not have to match. When there are fewer values than parameter names, either the RANDOM= or START= option is used. When there are more values than parameter names, the extra values are dropped. Parameters listed in the PARAMETERS statement can be assigned initial values by program statements or by the START= or RANDOM= option in the PROC CALIS statement.

CAUTION: The OUTRAM= and INRAM= data sets do not contain any information about the PARAMETERS statement or additional program statements.

STRUCTEQ Statement

STRUCTEQ variable < variable . . . > ;

The STRUCTEQ statement is used to list the dependent variables of the structural equations. This statement is ignored if you omit the PDETERM option. This statement is useful because the term structural equation is not defined in a unique way, and PROC CALIS has difficulty identifying the structural equations.

If LINEQS statements are used, the names of the left-hand-side (dependent) variables of those equations to be treated as structural equations should be listed in the STRUCTEQ statement.

If the RAM statement is used, variable names in the STRUCTEQ statements depend on the VARNAMES statement:

- If the VARNAMES statement is used, variable names must correspond to those in the VARNAMES statement.
- If the VARNAMES statement is not used, variable names must correspond to the names of manifest variables or latent (F) variables.

The STRUCTEQ statement also defines the names of variables used in the causal coefficient matrix of the structural equations, B, for computing the Stability Coefficient of Reciprocal Causation (the largest eigenvalue of the $BB'$ matrix). If the PROC CALIS option PDETERM is used without the STRUCTEQ statement, the structural equations are defined as described in the PDETERM option. See the PROC CALIS option PDETERM on page 432 for more details.

**VARNAMES Statement**

```
VARNAMES | VNAMES assignment < , assignment ... > ;
```

where assignment represents

```
matrix-id variable-names   or   matrix-name = matrix-name
```

Use the VARNAMES statement in connection with the RAM, COSAN, or FACTOR model statement to allocate names to latent variables including error and disturbance terms. This statement is not needed if you are using the LINEQS statement.

In connection with the RAM model statement, the matrix-id must be specified by the integer number as it is used in the RAM list input (1 for matrix A, 2 for matrix P). Because the first variables of matrix A correspond to the manifest variables in the input data set, you can specify names only for the latent variables following the manifest variables in the rows of A. For example, in the RAM notation of the alienation example, you can specify the latent variables by names F1, F2, F3 and the error variables by names E1, ..., E6, D1, D2, D3 with the following statement:

```
VARNAMES 1 F1-F3,
2 E1-E6 D1-D3;
```

If the RAM model statement is not accompanied by a VARNAMES statement, default variable names are assigned using the prefixes F, E, and D with numerical suffixes: latent variables are F1, F2, ..., and error variables are E1, E2, ....

The matrix-id must be specified by its name when used with the COSAN or FACTOR statement. The variable-names following the matrix name correspond to the columns of this matrix. The variable names corresponding to the rows of this matrix are set automatically by

- the names of the manifest variables for the first matrix in each term
- the column variable names of the same matrix for the central symmetric matrix in each term
- the column variable names of the preceding matrix for each other matrix
You also can use the second kind of name assignment in connection with a COSAN statement. Two matrix names separated by an equal sign allocate the column names of one matrix to the column names of the other matrix. This assignment assumes that the column names of at least one of the two matrices are already allocated. For example, in the COSAN notation of the alienation example, you can specify the variable names by using the following statements to allocate names to the columns of J, A, and P:

```
VNAMEs J V1-V6 F1-F3 ,
A = J ,
P E1-E6 D1-D3 ;
```

**BY Statement**

```
BY variables ;
```

You can specify a BY statement with PROC CALIS to obtain separate analyses on observations in groups defined by the BY variables. When a BY statement appears, the procedure expects the input data set to be sorted in order of the BY variables. If your input data set is not sorted in ascending order, use one of the following alternatives:

- Sort the data using the SORT procedure with a similar BY statement.
- Specify the BY statement option NOTSORTED or DESCENDING in the BY statement for the CALIS procedure. The NOTSORTED option does not mean that the data are unsorted but rather that the data are arranged in groups (according to values of the BY variables) and that these groups are not necessarily in alphabetical or increasing numeric order.
- Create an index on the BY variables using the DATASETS procedure.

For more information on the BY statement, refer to the discussion in *SAS Language Reference: Concepts*. For more information on the DATASETS procedure, refer to the discussion in the *SAS Procedures Guide*.

**VAR Statement**

```
VAR variables ;
```

The VAR statement lists the numeric variables to be analyzed. If the VAR statement is omitted, all numeric variables not mentioned in other statements are analyzed. You can use the VAR statement to ensure that the manifest variables appear in correct order for use in the RAM statement. Only one VAR statement can be used with each
PROC CALIS statement. If you do not use all manifest variables when you specify the model with a RAM or LINEQS statement, PROC CALIS does automatic variable selection. For more information, see the section “Automatic Variable Selection” on page 507.

PARTIAL Statement

PARTIAL variables ;

If you want the analysis to be based on a partial correlation or covariance matrix, use the PARTIAL statement to list the variables used to partial out the variables in the analysis. You can specify only one PARTIAL statement with each PROC CALIS statement.

FREQ Statement

FREQ variable ;

If one variable in your data set represents the frequency of occurrence for the other values in the observation, specify the variable’s name in a FREQ statement. PROC CALIS then treats the data set as if each observation appears \( n_i \) times, where \( n_i \) is the value of the FREQ variable for observation \( i \). Only the integer portion of the value is used. If the value of the FREQ variable is less than 1 or is missing, that observation is not included in the analysis. The total number of observations is considered to be the sum of the FREQ values when the procedure computes significance probabilities. You can use only one FREQ statement with each PROC CALIS statement.

WEIGHT Statement

WEIGHT variable ;

To compute weighted covariances or correlations, specify the name of the weighting variable in a WEIGHT statement. This is often done when the error variance associated with each observation is different and the values of the weight variable are proportional to the reciprocals of the variances. You can use only one WEIGHT statement with each PROC CALIS statement. The WEIGHT and FREQ statements have a similar effect, except the WEIGHT statement does not alter the number of observations unless VARDEF=WGT or VARDEF=WDF. An observation is used in the analysis only if the WEIGHT variable is greater than 0 and is not missing.
SAS Program Statements

This section lists the program statements used to express the linear and nonlinear constraints on the parameters and documents the differences between program statements in PROC CALIS and program statements in the DATA step. The very different use of the ARRAY statement by PROC CALIS is also discussed. Most of the program statements that can be used in the SAS DATA step also can be used in PROC CALIS. Refer to SAS Language Reference: Dictionary for a description of the SAS program statements. You can specify the following SAS program statements to compute parameter constraints with the CALIS procedure:

ABORT ;
CALL name < ( expression < , expression . . . > ) > ;
DELETE;
DO < variable = expression < TO expression > < BY expression >
   < , expression < TO expression > < BY expression > . . . > >
   < WHILE expression >
   < UNTIL expression > ;
END;
GOTO statement-label ;
IF expression ;
IF expression THEN program-statement ;
   ELSE program-statement ;
   variable = expression ;
   variable+expression ;
LINK statement-label ;
PUT <variable> <= < . . . > ;
RETURN ;
SELECT < ( expression ) > ;
STOP;
SUBSTR ( variable, index, length ) = expression ;
WHEN (expression) program-statement ;
   OTHERWISE program-statement ;

For the most part, the SAS program statements work the same as they do in the SAS DATA step as documented in SAS Language Reference: Concepts. However, there are several differences that should be noted.

- The ABORT statement does not allow any arguments.
- The DO statement does not allow a character index variable. Thus,

  DO I=1,2,3;

  is supported; however,
  DO I=’A’,’B’,’C’ ;
is not valid in PROC CALIS, although it is supported in the DATA step.

- The PUT statement, used mostly for program debugging in PROC CALIS, supports only some of the features of the DATA step PUT statement, and it has some new features that the DATA step PUT statement does not have:
  
  - The CALIS procedure PUT statement does not support line pointers, factored lists, iteration factors, overprinting, _INFILE_, the colon (:): format modifier, or $.
  
  - The CALIS procedure PUT statement does support expressions enclosed in parentheses. For example, the following statement displays the square root of x:

    \[
    \text{PUT (SQRT(x))}
    \]

  - The CALIS procedure PUT statement supports the print item _PDV_ to display a formatted listing of all variables in the program. For example, the following statement displays a much more readable listing of the variables than the _ALL_ print item:

    \[
    \text{PUT _PDV_}
    \]

- The WHEN and OTHERWISE statements allow more than one target statement. That is, DO/END groups are not necessary for multiple WHEN statements. For example, the following syntax is valid:

    \[
    \text{SELECT;}
    \text{WHEN ( expression1 ) statement1;}
    \text{statement2;}
    \text{WHEN ( expression2 ) statement3;}
    \text{statement4;}
    \text{END;}
    \]

You can specify one or more PARMS statements to define parameters used in the program statements that are not defined in the model matrices (MATRIX, RAM, LINEQS, STD, or COV statement).

Parameters that are used only on the right-hand side of your program statements are called independent, and parameters that are used at least once on the left-hand side of an equation in the program code are called dependent parameters. The dependent parameters are used only indirectly in the minimization process. They should be fully defined as functions of the independent parameters. The independent parameters are included in the set \(X\) of parameters used in the minimization. Be sure that all independent parameters used in your program statements are somehow connected to elements of the model matrices. Otherwise the minimization function does not depend on those independent parameters, and the parameters vary without control (since the corresponding derivative is the constant 0). You also can specify the PARMS statement to set the initial values of all independent parameters used in the program statements that are not defined as elements of model matrices.
Chapter 17. The CALIS Procedure

**ARRAY Statement**

```
ARRAY arrayname <(dimensions)> $ <variables and constants> ;
```

The ARRAY statement is similar to, but not the same as, the ARRAY statement in the DATA step. The ARRAY statement is used to associate a name with a list of variables and constants. The array name can then be used with subscripts in the program to refer to the items in the list.

The ARRAY statement supported by PROC CALIS does not support all the features of the DATA step ARRAY statement. With PROC CALIS, the ARRAY statement cannot be used to give initial values to array elements. Implicit indexing variables cannot be used; all array references must have explicit subscript expressions. Only exact array dimensions are allowed; lower-bound specifications are not supported. A maximum of six dimensions is allowed.

On the other hand, the ARRAY statement supported by PROC CALIS does allow both variables and constants to be used as array elements. Constant array elements cannot be changed. Both the dimension specification and the list of elements are optional, but at least one must be given. When the list of elements is not given or fewer elements than the size of the array are listed, array variables are created by suffixing element numbers to the array name to complete the element list.

---

**Details**

**Input Data Sets**

You can use four different kinds of input data sets in the CALIS procedure, and you can use them simultaneously. The DATA= data set contains the data to be analyzed, and it can be an ordinary SAS data set containing raw data or a special TYPE=COV, TYPE=UCOV, TYPE=CORR, TYPE=UCORR, TYPE=SYMATRIX, TYPE=SSCP, or TYPE=FACTOR data set containing previously computed statistics. The INEST= data set specifies an input data set that contains initial estimates for the parameters used in the optimization process, and it can also contain boundary and general linear constraints on the parameters. If the model does not change too much, you can use an OUTEST= data set from a previous PROC CALIS analysis; the initial estimates are taken from the values of the PARMS observation. The INRAM= data set names a third input data set that contains all information needed to specify the analysis model in RAM list form (except for user-written program statements). Often the INRAM= data set can be the OUTRAM= data set from a previous PROC CALIS analysis. See the section “OUTRAM= SAS-data-set” on page 484 for the structure of both OUTRAM= and INRAM= data sets. Using the INWGT= data set enables you to read in the weight matrix \( W \) that can be used in generalized least-squares, weighted least-squares, or diagonally weighted least-squares estimation.
**DATA= SAS-data-set**

A TYPE=COV, TYPE=UCOV, TYPE=CORR, or TYPE=UCORR data set can be created by the CORR procedure or various other procedures. It contains means, standard deviations, the sample size, the covariance or correlation matrix, and possibly other statistics depending on which procedure is used.

If your data set has many observations and you plan to run PROC CALIS several times, you can save computer time by first creating a TYPE=COV, TYPE=UCOV, TYPE=CORR, or TYPE=UCORR data set and using it as input to PROC CALIS. For example, assuming that PROC CALIS is first run with an OUTRAM=MOD option, you can run

```sas
* create TYPE=COV data set;
PROC CORR COV NOCORR DATA=RAW OUTP=COV(TYPE=COV);
RUN;
* analysis using correlations;
PROC CALIS DATA=COV INRAM=MOD;
RUN;
* analysis using covariances;
PROC CALIS COV DATA=COV INRAM=MOD;
RUN;
```

Most procedures automatically set the TYPE= option of an output data set appropriately. However, the CORR procedure sets TYPE=CORR unless an explicit TYPE= option is used. Thus, (TYPE=COV) is needed in the preceding PROC CORR request, since the output data set is a covariance matrix. If you use a DATA step with a SET statement to modify this data set, you must declare the TYPE=COV, TYPE=UCOV, TYPE=CORR, or TYPE=UCORR attribute in the new data set.

You can use a VAR statement with PROC CALIS when reading a TYPE=COV, TYPE=UCOV, TYPE=CORR, TYPE=UCORR, or TYPE=SSCP data set to select a subset of the variables or change the order of the variables.

**CAUTION:** Problems can arise from using the CORR procedure when there are missing data. By default, PROC CORR computes each covariance or correlation from all observations that have values present for the pair of variables involved (“pairwise deletion”). The resulting covariance or correlation matrix can have negative eigenvalues. A correlation or covariance matrix with negative eigenvalues is recognized as a singular matrix in PROC CALIS, and you cannot compute (default) generalized least-squares or maximum likelihood estimates. You can specify the RIDGE option to ridge the diagonal of such a matrix to obtain a positive definite data matrix. If the NOMISS option is used with the CORR procedure, observations with any missing values are completely omitted from the calculations (“listwise deletion”), and there is no possibility of negative eigenvalues (but still a chance for a singular matrix).

PROC CALIS can also create a TYPE=COV, TYPE=UCOV, TYPE=CORR, or TYPE=UCORR data set that includes all the information needed for repeated analyses. If the data set DATA=RAW does not contain missing values, the following statements should give the same PROC CALIS results as the previous example:
* using correlations;
PROC CALIS DATA=RAW OUTSTAT=COV INRAM=MOD;
RUN;
* using covariances;
PROC CALIS COV DATA=COV INRAM=MOD;
RUN;

You can create a TYPE=COV, TYPE=UCOV, TYPE=CORR, TYPE=UCORR, or TYPE=SSCP data set in a DATA step. Be sure to specify the TYPE= option in parentheses after the data set name in the DATA statement, and include the _TYPE_ and _NAME_ variables. If you want to analyze the covariance matrix but your DATA= data set is a TYPE=CORR or TYPE=UCORR data set, you should include an observation with _TYPE_=STD giving the standard deviation of each variable. If you specify the COV option, PROC CALIS analyzes the recomputed covariance matrix:

```
DATA CORREL(TYPE=CORR);
   INPUT _TYPE_ $ _NAME_ $ X1-X3;
DATALINES;
   STD . 4. 2. 8.
   CORR X1 1.0 . .
   CORR X2 .7 1.0 .
   CORR X3 .5 .4 1.0
;
PROC CALIS COV INRAM=MODEL;
RUN;
```

If you want to analyze the UCOV or UCORR matrix but your DATA= data set is a TYPE=COV or TYPE=CORR data set, you should include observations with _TYPE_=STD and _TYPE_=MEAN giving the standard deviation and mean of each variable.

**INEST= SAS-data-set**

You can use the INEST= (or INVAR= or ESTDATA=) input data set to specify the initial values of the parameters used in the optimization and to specify boundary constraints and the more general linear constraints that can be imposed on these parameters.

The variables of the INEST= data set must correspond to

- a character variable _TYPE_ that indicates the type of the observation
- n numeric variables with the parameter names used in the specified PROC CALIS model
- the BY variables that are used in a DATA= input data set
- a numeric variable _RHS_ (right-hand side) (needed only if linear constraints are used)
- additional variables with names corresponding to constants used in the program statements
The content of the _TYPE_ variable defines the meaning of the observation of the INEST= data set. PROC CALIS recognizes observations with the following _TYPE_ specifications.

PARMS specifies initial values for parameters that are defined in the model statements of PROC CALIS. The _RHS_ variable is not used. Additional variables can contain the values of constants that are referred to in program statements. At the beginning of each run of PROC CALIS, the values of the constants are read from the PARMS observation initializing the constants in the program statements.

UPPERBD | UB specifies upper bounds with nonmissing values. The use of a missing value indicates that no upper bound is specified for the parameter. The _RHS_ variable is not used.

LOWERBD | LB specifies lower bounds with nonmissing values. The use of a missing value indicates that no lower bound is specified for the parameter. The _RHS_ variable is not used.

LE | <= | < specifies the linear constraint \( \sum_j a_{ij}x_j \leq b_i \). The \( n \) parameter values contain the coefficients \( a_{ij} \), and the _RHS_ variable contains the right-hand-side \( b_i \). The use of a missing value indicates a zero coefficient \( a_{ij} \).

GE | >= | > specifies the linear constraint \( \sum_j a_{ij}x_j \geq b_i \). The \( n \) parameter values contain the coefficients \( a_{ij} \), and the _RHS_ variable contains the right-hand-side \( b_i \). The use of a missing value indicates a zero coefficient \( a_{ij} \).

EQ | = specifies the linear constraint \( \sum_j a_{ij}x_j = b_i \). The \( n \) parameter values contain the coefficients \( a_{ij} \), and the _RHS_ variable contains the right-hand-side \( b_i \). The use of a missing value indicates a zero coefficient \( a_{ij} \).

The constraints specified in the INEST=, INVAR=, or ESTDATA= data set are added to the constraints specified in BOUNDS and LINCON statements.

You can use an OUTEST= data set from a PROC CALIS run as an INEST= data set in a new run. However, be aware that the OUTEST= data set also contains the boundary and general linear constraints specified in the previous run of PROC CALIS. When you are using this OUTEST= data set without changes as an INEST= data set, PROC CALIS adds the constraints from the data set to the constraints specified by a BOUNDS and LINCON statement. Although PROC CALIS automatically eliminates multiple identical constraints, you should avoid specifying the same constraint a second time.

**INRAM= SAS-data-set**

This data set is usually created in a previous run of PROC CALIS. It is useful if you want to reanalyze a problem in a different way such as using a different estimation method. You can alter an existing OUTRAM= data set, either in the DATA step or using the FSEDIT procedure, to create the INRAM= data set describing a modified
In the case of a RAM or LINEQS analysis of linear structural equations, the OUTRAM= data set always contains the variable names of the model specified. These variable names and the model specified in the INRAM= data set are the basis of the automatic variable selection algorithm performed after reading the INRAM= data set.

**INWGT= SAS-data-set**

This data set enables you to specify a weight matrix other than the default matrix for the generalized, weighted, and diagonally weighted least-squares estimation methods. The specification of any INWGT= data set for unweighted least-squares or maximum likelihood estimation is ignored. For generalized and diagonally weighted least-squares estimation, the INWGT= data set must contain a _TYPE_ and a _NAME_ variable as well as the manifest variables used in the analysis. The value of the _NAME_ variable indicates the row index \( i \) of the weight \( w_{ij} \). For weighted least squares, the INWGT= data set must contain _TYPE_, _NAME_, _NAM2_, and _NAM3_ variables as well as the manifest variables used in the analysis. The values of the _NAME_, _NAM2_, and _NAM3_ variables indicate the three indices \( i, j, k \) of the weight \( w_{ij,k} \). You can store information other than the weight matrix in the INWGT= data set, but only observations with _TYPE_=WEIGHT are used to specify the weight matrix \( W \). This property enables you to store more than one weight matrix in the INWGT= data set. You can then run PROC CALIS with each of the weight matrices by changing only the _TYPE_ observation in the INWGT= data set with an intermediate DATA step.

For more details on the INWGT= data set, see the section “OUTWGT= SAS-data-set” on page 489.

---

**Output Data Sets**

**OUTEST= SAS-data-set**

The OUTEST= (or OUTVAR=) data set is of TYPE=EST and contains the final parameter estimates, the gradient, the Hessian, and boundary and linear constraints. For METHOD=ML, METHOD=GLS, and METHOD=WLS, the OUTEST= data set also contains the approximate standard errors, the information matrix (crossproduct Jacobian), and the approximate covariance matrix of the parameter estimates ((generalized) inverse of the information matrix). If there are linear or nonlinear equality or active inequality constraints at the solution, the OUTEST= data set also contains Lagrange multipliers, the projected Hessian matrix, and the Hessian matrix of the Lagrange function.

The OUTEST= data set can be used to save the results of an optimization by PROC CALIS for another analysis with either PROC CALIS or another SAS procedure. Saving results to an OUTEST= data set is advised for expensive applications that cannot be repeated without considerable effort.

The OUTEST= data set contains the BY variables, two character variables _TYPE_ and _NAME_, \( t \) numeric variables corresponding to the parameters used in the model,
a numeric variable `_RHS_` (right-hand side) that is used for the right-hand-side value \( b_i \) of a linear constraint or for the value \( f = f(x) \) of the objective function at the final point \( x^* \) of the parameter space, and a numeric variable `_ITER_` that is set to zero for initial values, set to the iteration number for the OUTITER output, and set to missing for the result output.

The `_TYPE_` observations in Table 17.5 are available in the OUTEST= data set, depending on the request.

**Table 17.5. `_TYPE_` Observations in the OUTEST= data set**

<table>
<thead>
<tr>
<th><code>_TYPE_</code></th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ACTBC</td>
<td>If there are active boundary constraints at the solution ( x^* ), three observations indicate which of the parameters are actively constrained, as follows.</td>
</tr>
<tr>
<td>GE</td>
<td>indicates the active lower bounds</td>
</tr>
<tr>
<td>LE</td>
<td>indicates the active upper bounds</td>
</tr>
<tr>
<td>EQ</td>
<td>indicates the active masks</td>
</tr>
<tr>
<td>COV</td>
<td>contains the approximate covariance matrix of the parameter estimates; used in computing the approximate standard errors.</td>
</tr>
<tr>
<td>COVRANK</td>
<td>contains the rank of the covariance matrix of the parameter estimates.</td>
</tr>
<tr>
<td>CRPJ_LF</td>
<td>contains the Hessian matrix of the Lagrange function (based on CRPJAC).</td>
</tr>
<tr>
<td>CRPJAC</td>
<td>contains the approximate Hessian matrix used in the optimization process. This is the inverse of the information matrix.</td>
</tr>
<tr>
<td>EQ</td>
<td>If linear constraints are used, this observation contains the ( i )th linear constraint ( \sum_j a_{ij} x_j = b_i ). The parameter variables contain the coefficients ( a_{ij}, \ j = 1, \ldots, n ), the <code>_RHS_</code> variable contains ( b_i ), and <code>_NAME_=ACTLC</code> or <code>_NAME_=LDACTLC</code>.</td>
</tr>
<tr>
<td>GE</td>
<td>If linear constraints are used, this observation contains the ( i )th linear constraint ( \sum_j a_{ij} x_j \geq b_i ). The parameter variables contain the coefficients ( a_{ij}, \ j = 1, \ldots, n ), and the <code>_RHS_</code> variable contains ( b_i ). If the constraint ( i ) is active at the solution ( x^* ), then <code>_NAME_=ACTLC</code> or <code>_NAME_=LDACTLC</code>.</td>
</tr>
<tr>
<td>GRAD</td>
<td>contains the gradient of the estimates.</td>
</tr>
<tr>
<td>GRAD_LF</td>
<td>contains the gradient of the Lagrange function. The <code>_RHS_</code> variable contains the value of the Lagrange function.</td>
</tr>
<tr>
<td>HESSIAN</td>
<td>contains the Hessian matrix.</td>
</tr>
<tr>
<td>HESS_LF</td>
<td>contains the Hessian matrix of the Lagrange function (based on HESSIAN).</td>
</tr>
</tbody>
</table>
Table 17.5. _TYPE_ Observations in the OUTEST= data set (continued)

<table>
<thead>
<tr>
<th><em>TYPE</em></th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>INFORMAT</td>
<td>contains the information matrix of the parameter estimates (only for METHOD=ML, METHOD=GLS, or METHOD=WLS).</td>
</tr>
<tr>
<td>INITIAL</td>
<td>contains the starting values of the parameter estimates.</td>
</tr>
<tr>
<td>JACNLC</td>
<td>contains the Jacobian of the nonlinear constraints evaluated at the final estimates.</td>
</tr>
<tr>
<td>JACOBIAN</td>
<td>contains the Jacobian matrix (only if the OUTJAC option is used).</td>
</tr>
<tr>
<td>LAGM BC</td>
<td>contains Lagrange multipliers for masks and active boundary constraints.</td>
</tr>
<tr>
<td><em>NAME</em></td>
<td>Description</td>
</tr>
<tr>
<td></td>
<td>GE</td>
</tr>
<tr>
<td></td>
<td>LE</td>
</tr>
<tr>
<td></td>
<td>EQ</td>
</tr>
<tr>
<td>LAGM LC</td>
<td>contains Lagrange multipliers for linear equality and active inequality constraints in pairs of observations containing the constraint number and the value of the Lagrange multiplier.</td>
</tr>
<tr>
<td><em>NAME</em></td>
<td>Description</td>
</tr>
<tr>
<td></td>
<td>LEC.NUM</td>
</tr>
<tr>
<td></td>
<td>LEC.VAL</td>
</tr>
<tr>
<td></td>
<td>LIC.NUM</td>
</tr>
<tr>
<td></td>
<td>LIC.VAL</td>
</tr>
<tr>
<td>LAGM NLC</td>
<td>contains Lagrange multipliers for nonlinear equality and active inequality constraints in pairs of observations containing the constraint number and the value of the Lagrange multiplier.</td>
</tr>
<tr>
<td><em>NAME</em></td>
<td>Description</td>
</tr>
<tr>
<td></td>
<td>NLEC.NUM</td>
</tr>
<tr>
<td></td>
<td>NLEC.VAL</td>
</tr>
<tr>
<td></td>
<td>NLIC.NUM</td>
</tr>
<tr>
<td></td>
<td>NLIC.VAL</td>
</tr>
<tr>
<td>LE</td>
<td>If linear constraints are used, this observation contains the $i$th linear constraint $\sum_j a_{ij}x_j \leq b_i$. The parameter variables contain the coefficients $a_{ij}$, $j = 1, \ldots, n$, and the <em>RHS</em> variable contains $b_i$. If the constraint $i$ is active at the solution $x^*$, then <em>NAME</em>=ACTLC or <em>NAME</em>=LDACTLC.</td>
</tr>
<tr>
<td>LOWERBD</td>
<td>If boundary constraints are used, this observation contains the lower bounds. Those parameters not subjected to lower bounds contain missing values. The <em>RHS</em> variable contains a missing value, and the <em>NAME</em> variable is blank.</td>
</tr>
<tr>
<td>LB</td>
<td></td>
</tr>
</tbody>
</table>
### Table 17.5. _TYPE_ Observations in the OUTEST= data set (continued)

<table>
<thead>
<tr>
<th><em>TYPE</em></th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>NACTBC</td>
<td>All parameter variables contain the number $n_{aLC}$ of active boundary constraints at the solution $x^*$. The <em>RHS</em> variable contains a missing value, and the <em>NAME</em> variable is blank.</td>
</tr>
<tr>
<td>NACTLC</td>
<td>All parameter variables contain the number $n_{aLC}$ of active linear constraints at the solution $x^*$ that are recognized as linearly independent. The <em>RHS</em> variable contains a missing value, and the <em>NAME</em> variable is blank.</td>
</tr>
<tr>
<td>NLC_EQ</td>
<td>contains values and residuals of nonlinear constraints. The <em>NAME</em> variable is described as follows.</td>
</tr>
<tr>
<td>NLC_GE</td>
<td><em>NAME</em> inactive nonlinear constraint</td>
</tr>
<tr>
<td>NLC_LE</td>
<td><em>NAME</em> linear independent active nonlinear constr.</td>
</tr>
<tr>
<td></td>
<td><em>NAME</em> linear dependent active nonlinear constr.</td>
</tr>
<tr>
<td>NLDACTBC</td>
<td>contains the number of active boundary constraints at the solution $x^*$ that are recognized as linearly dependent. The <em>RHS</em> variable contains a missing value, and the <em>NAME</em> variable is blank.</td>
</tr>
<tr>
<td>NLDACTLC</td>
<td>contains the number of active linear constraints at the solution $x^*$ that are recognized as linearly dependent. The <em>RHS</em> variable contains a missing value, and the <em>NAME</em> variable is blank.</td>
</tr>
<tr>
<td><em>NOBS</em></td>
<td>contains the number of observations.</td>
</tr>
<tr>
<td>PARMS</td>
<td>contains the final parameter estimates. The <em>RHS</em> variable contains the value of the objective function.</td>
</tr>
<tr>
<td>PCRPJ_LF</td>
<td>contains the projected Hessian matrix of the Lagrange function (based on CRPJAC).</td>
</tr>
<tr>
<td>PHESS_LF</td>
<td>contains the projected Hessian matrix of the Lagrange function (based on HESSIAN).</td>
</tr>
<tr>
<td>PROJCRPJ</td>
<td>contains the projected Hessian matrix (based on CRPJAC).</td>
</tr>
<tr>
<td>PROJGRAD</td>
<td>If linear constraints are used in the estimation, this observation contains the $n - n_{act}$ values of the projected gradient $g = Z'g$ in the variables corresponding to the first $n - n_{act}$ parameters. The <em>RHS</em> variable contains a missing value, and the <em>NAME</em> variable is blank.</td>
</tr>
<tr>
<td>PROJHESS</td>
<td>contains the projected Hessian matrix (based on HESSIAN).</td>
</tr>
<tr>
<td>SIGSQ</td>
<td>contains the scalar factor of the covariance matrix of the parameter estimates.</td>
</tr>
<tr>
<td>STDERR</td>
<td>contains approximate standard errors (only for METHOD=ML, METHOD=GLS, or METHOD=WLS).</td>
</tr>
</tbody>
</table>
Table 17.5.  _TYPE_ Observations in the OUTEST= data set (continued)

<table>
<thead>
<tr>
<th><em>TYPE</em></th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>TERMINAT</td>
<td>The <em>NAME</em> variable contains the name of the termination criterion.</td>
</tr>
<tr>
<td>UPPERBD</td>
<td>If boundary constraints are used, this observation contains the upper bounds. Those parameters not subjected to upper bounds contain missing values. The <em>RHS</em> variable contains a missing value, and the <em>NAME</em> variable is blank.</td>
</tr>
</tbody>
</table>

If the technique specified by the TECH= option cannot be performed (for example, no feasible initial values can be computed, or the function value or derivatives cannot be evaluated at the starting point), the OUTEST= data set may contain only some of the observations (usually only the PARMS and GRAD observations).

**OUTRAM= SAS-data-set**

The OUTRAM= data set is of TYPE=RAM and contains the model specification and the computed parameter estimates. This data set is intended to be reused as an INRAM= data set to specify good initial values in a subsequent analysis by PROC CALIS. For a structural equation model, after some alterations, this data set can be used for plotting a path diagram by the NETDRAW procedure, a SAS/OR procedure.

The OUTRAM= data set contains the following variables:

- the BY variables, if any
- the character variable _TYPE_, which takes the values MODEL, ESTIM, VARNAME, METHOD, and STAT
- six additional variables whose meaning depends on the _TYPE_ of the observation

Each observation with _TYPE_ =MODEL defines one matrix in the generalized COSAN model. The additional variables are as follows.

Table 17.6.  Additional Variables when _TYPE_ =MODEL

<table>
<thead>
<tr>
<th>Variable</th>
<th>Contents</th>
</tr>
</thead>
<tbody>
<tr>
<td><em>NAME</em></td>
<td>name of the matrix (character)</td>
</tr>
<tr>
<td><em>MATNR</em></td>
<td>number of term and matrix in the model (numeric)</td>
</tr>
<tr>
<td><em>ROW</em></td>
<td>matrix row number (numeric)</td>
</tr>
<tr>
<td><em>COL</em></td>
<td>matrix column number (numeric)</td>
</tr>
<tr>
<td><em>ESTIM</em></td>
<td>first matrix type (numeric)</td>
</tr>
<tr>
<td>STDERR_</td>
<td>second matrix type (numeric)</td>
</tr>
</tbody>
</table>

If the generalized COSAN model has only one matrix term, the _MATNR_ variable contains only the number of the matrix in the term. If there is more than one term, then it is the term number multiplied by 10,000 plus the matrix number (assuming that there are no more than 9,999 matrices specified in the COSAN model statement).
Each observation with \_TYPE\_ =ESTIM defines one element of a matrix in the generalized COSAN model. The variables are used as follows.

**Table 17.7. Additional Variables when \_TYPE\_=ESTIM**

<table>
<thead>
<tr>
<th>Variable</th>
<th>Contents</th>
</tr>
</thead>
<tbody>
<tr>
<td>_NAME_</td>
<td>name of the parameter (character)</td>
</tr>
<tr>
<td>_MATNR_</td>
<td>term and matrix location of parameter (numeric)</td>
</tr>
<tr>
<td>_ROW_</td>
<td>row location of parameter (numeric)</td>
</tr>
<tr>
<td>_COL_</td>
<td>column location of parameter (numeric)</td>
</tr>
<tr>
<td>_ESTIM_</td>
<td>parameter estimate or constant value (numeric)</td>
</tr>
<tr>
<td>_STDERR_</td>
<td>standard error of estimate (numeric)</td>
</tr>
</tbody>
</table>

For constants rather than estimates, the \_STDERR\_ variable is 0. The \_STDERR\_ variable is missing for ULS and DWLS estimates if NOSTDERR is specified or if the approximate standard errors are not computed.

Each observation with \_TYPE\_ =VARNAME defines a column variable name of a matrix in the generalized COSAN model.

The observations with \_TYPE\_ =METHOD and \_TYPE\_ =STAT are not used to build the model. The \_TYPE\_ =METHOD observation contains the name of the estimation method used to compute the parameter estimates in the \_NAME\_ variable. If METHOD=NONE is not specified, the \_ESTIM\_ variable of the \_TYPE\_ =STAT observations contains the information summarized in Table 17.8 (described in the section “Assessment of Fit” on page 495).

**Table 17.8. \_ESTIM\_ Contents for \_TYPE\_ =STAT**

<table>
<thead>
<tr>
<th>_NAME_</th>
<th>_ESTIM_</th>
</tr>
</thead>
<tbody>
<tr>
<td>N</td>
<td>sample size</td>
</tr>
<tr>
<td>NPARM</td>
<td>number of parameters used in the model</td>
</tr>
<tr>
<td>DF</td>
<td>degrees of freedom</td>
</tr>
<tr>
<td>N_ACT</td>
<td>number of active boundary constraints for ML, GLS, and WLS estimation</td>
</tr>
<tr>
<td>FIT</td>
<td>fit function</td>
</tr>
<tr>
<td>GFI</td>
<td>goodness-of-fit index (GFI)</td>
</tr>
<tr>
<td>AGFI</td>
<td>adjusted GFI for degrees of freedom</td>
</tr>
<tr>
<td>RMR</td>
<td>root mean square residual</td>
</tr>
<tr>
<td>PGFI</td>
<td>parsimonious GFI of Mulaik et al. (1989)</td>
</tr>
<tr>
<td>CHISQUAR</td>
<td>overall (\chi^2)</td>
</tr>
<tr>
<td>P_CHISQ</td>
<td>probability &gt; (\chi^2)</td>
</tr>
<tr>
<td>CHISQNUL</td>
<td>null (baseline) model (\chi^2)</td>
</tr>
<tr>
<td>RMSEAEST</td>
<td>Steiger &amp; Lind’s (1980) RMSEA index estimate</td>
</tr>
<tr>
<td>RMSEALOB</td>
<td>lower range of RMSEA confidence interval</td>
</tr>
<tr>
<td>RMSEAUPB</td>
<td>upper range of RMSEA confidence interval</td>
</tr>
<tr>
<td>P_CLOSFT</td>
<td>Browne &amp; Cudeck’s (1993) probability of close fit</td>
</tr>
<tr>
<td>ECVI_EST</td>
<td>Browne &amp; Cudeck’s (1993) ECV index estimate</td>
</tr>
<tr>
<td>ECVI_LOB</td>
<td>lower range of ECVI confidence interval</td>
</tr>
<tr>
<td>ECVI_UPB</td>
<td>upper range of ECVI confidence interval</td>
</tr>
<tr>
<td>COMPFITI</td>
<td>Bentler’s (1989) comparative fit index</td>
</tr>
</tbody>
</table>
Table 17.8. _ESTIM_ Contents for _TYPE_=STAT (continued)

<table>
<thead>
<tr>
<th><em>NAME</em></th>
<th><em>ESTIM</em></th>
</tr>
</thead>
<tbody>
<tr>
<td>ADJCHISQ</td>
<td>adjusted $\chi^2$ for elliptic distribution</td>
</tr>
<tr>
<td>P_ACHISQ</td>
<td>probability corresponding adjusted $\chi^2$</td>
</tr>
<tr>
<td>RLSCHISQ</td>
<td>reweighted least-squares $\chi^2$ (only ML estimation)</td>
</tr>
<tr>
<td>AIC</td>
<td>Akaike’s information criterion</td>
</tr>
<tr>
<td>CAIC</td>
<td>Bozdogan’s consistent information criterion</td>
</tr>
<tr>
<td>SBC</td>
<td>Schwarz’s Bayesian criterion</td>
</tr>
<tr>
<td>CENTRALI</td>
<td>McDonald’s centrality criterion</td>
</tr>
<tr>
<td>PARSIMON</td>
<td>Parsimonious index of James, Mulaik, and Brett</td>
</tr>
<tr>
<td>ZTESTWH</td>
<td>$z$ test of Wilson and Hilferty</td>
</tr>
<tr>
<td>BB_NONOR</td>
<td>Bentler-Bonett (1980) nonnormed index $\rho$</td>
</tr>
<tr>
<td>BB_NORMD</td>
<td>Bentler-Bonett (1980) normed index $\Delta$</td>
</tr>
<tr>
<td>BOL_RHO1</td>
<td>Bollen’s (1986) normed index $\rho_1$</td>
</tr>
<tr>
<td>BOL_DEL2</td>
<td>Bollen’s (1989a) nonnormed index $\Delta_2$</td>
</tr>
<tr>
<td>CNHOELT</td>
<td>Hoelter’s critical N index</td>
</tr>
</tbody>
</table>

You can edit the OUTRAM= data set to use its contents for initial estimates in a subsequent analysis by PROC CALIS, perhaps with a slightly changed model. But you should be especially careful for _TYPE_=MODEL when changing matrix types. The codes for the two matrix types are listed in Table 17.9.

Table 17.9. Matrix Type Codes

<table>
<thead>
<tr>
<th>Code</th>
<th>First Matrix Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1:</td>
<td>IDE</td>
<td>identity matrix</td>
</tr>
<tr>
<td>2:</td>
<td>ZID</td>
<td>zero:identity matrix</td>
</tr>
<tr>
<td>3:</td>
<td>DIA</td>
<td>diagonal matrix</td>
</tr>
<tr>
<td>4:</td>
<td>ZDI</td>
<td>zero:diagonal matrix</td>
</tr>
<tr>
<td>5:</td>
<td>LOW</td>
<td>lower triangular matrix</td>
</tr>
<tr>
<td>6:</td>
<td>UPP</td>
<td>upper triangular matrix</td>
</tr>
<tr>
<td>7:</td>
<td>T</td>
<td>temporarily not used</td>
</tr>
<tr>
<td>8:</td>
<td>SYM</td>
<td>symmetric matrix</td>
</tr>
<tr>
<td>9:</td>
<td>GEN</td>
<td>general-type matrix</td>
</tr>
<tr>
<td>10:</td>
<td>BET</td>
<td>identity minus general-type matrix</td>
</tr>
<tr>
<td>11:</td>
<td>PER</td>
<td>selection matrix</td>
</tr>
<tr>
<td>12:</td>
<td></td>
<td>first matrix (J) in LINEQS model statement</td>
</tr>
<tr>
<td>13:</td>
<td></td>
<td>second matrix ((\beta)) in LINEQS model statement</td>
</tr>
<tr>
<td>14:</td>
<td></td>
<td>third matrix ((\gamma)) in LINEQS model statement</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Code</th>
<th>Second Matrix Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0:</td>
<td>NON</td>
<td>noninverse model matrix</td>
</tr>
<tr>
<td>1:</td>
<td>INV</td>
<td>inverse model matrix</td>
</tr>
<tr>
<td>2:</td>
<td>IMI</td>
<td>‘identity minus inverse’ model matrix</td>
</tr>
</tbody>
</table>
### OUTSTAT= SAS-data-set

The OUTSTAT= data set is similar to the TYPE=COV, TYPE=UCOV, TYPE=CORR, or TYPE=UCORR data set produced by the CORR procedure. The OUTSTAT= data set contains the following variables:

- the BY variables, if any
- two character variables, _TYPE_ and _NAME_
- the variables analyzed, that is, those in the VAR statement, or if there is no VAR statement, all numeric variables not listed in any other statement but used in the analysis. (CAUTION: Using the LINEQS or RAM model statements selects variables automatically.)

The OUTSTAT= data set contains the following information (when available):

- the mean and standard deviation
- the skewness and kurtosis (if the DATA= data set is a raw data set and the KURTOSIS option is specified)
- the number of observations
- if the WEIGHT statement is used, sum of the weights
- the correlation or covariance matrix to be analyzed
- the predicted correlation or covariance matrix
- the standardized or normalized residual correlation or covariance matrix
- if the model contains latent variables, the predicted covariances between latent and manifest variables
- if the FACTOR statement is used,
  - the unrotated factor loadings, the unique variances, and the matrix of factor correlations
  - the rotated factor loadings and the transformation matrix of the rotation
  - the matrix of standardized factor loadings
  - latent variable score regression coefficients (see the PLATCOV display option on page 432)

Each observation in the OUTSTAT= data set contains some type of statistic as indicated by the _TYPE_ variable. The values of the _TYPE_ variable are given in Table 17.10.
Table 17.10. _TYPE_ Observations in the OUTSTAT= data set

<table>
<thead>
<tr>
<th><em>TYPE</em></th>
<th>Contents</th>
</tr>
</thead>
<tbody>
<tr>
<td>MEAN</td>
<td>means</td>
</tr>
<tr>
<td>STD</td>
<td>standard deviations</td>
</tr>
<tr>
<td>USTD</td>
<td>uncorrected standard deviations</td>
</tr>
<tr>
<td>SKEWNESS</td>
<td>univariate skewness</td>
</tr>
<tr>
<td>KURTOSIS</td>
<td>univariate kurtosis</td>
</tr>
<tr>
<td>N</td>
<td>sample size</td>
</tr>
<tr>
<td>SUMWGT</td>
<td>sum of weights (if WEIGHT statement is used)</td>
</tr>
<tr>
<td>COV</td>
<td>covariances analyzed</td>
</tr>
<tr>
<td>CORR</td>
<td>correlations analyzed</td>
</tr>
<tr>
<td>UCOV</td>
<td>uncorrected covariances analyzed</td>
</tr>
<tr>
<td>UCORR</td>
<td>uncorrected correlations analyzed</td>
</tr>
<tr>
<td>ULSPRED</td>
<td>ULS predicted model values</td>
</tr>
<tr>
<td>GLSPRED</td>
<td>GLS predicted model values</td>
</tr>
<tr>
<td>MAXPRED</td>
<td>ML predicted model values</td>
</tr>
<tr>
<td>WLSNRES</td>
<td>WLS predicted model values</td>
</tr>
<tr>
<td>DWLSPRED</td>
<td>DWLS predicted model values</td>
</tr>
<tr>
<td>ULSNRES</td>
<td>ULS normalized residuals</td>
</tr>
<tr>
<td>GLSNRES</td>
<td>GLS normalized residuals</td>
</tr>
<tr>
<td>MAXNRES</td>
<td>ML normalized residuals</td>
</tr>
<tr>
<td>WLSNRES</td>
<td>WLS normalized residuals</td>
</tr>
<tr>
<td>DWLSNRES</td>
<td>DWLS normalized residuals</td>
</tr>
<tr>
<td>ULSSRES</td>
<td>ULS variance standardized residuals</td>
</tr>
<tr>
<td>GLSSRES</td>
<td>GLS variance standardized residuals</td>
</tr>
<tr>
<td>MAXSRES</td>
<td>ML variance standardized residuals</td>
</tr>
<tr>
<td>WLSSRES</td>
<td>WLS variance standardized residuals</td>
</tr>
<tr>
<td>DWLSSRES</td>
<td>DWLS variance standardized residuals</td>
</tr>
<tr>
<td>ULSASRES</td>
<td>ULS asymptotically standardized residuals</td>
</tr>
<tr>
<td>GLSASRES</td>
<td>GLS asymptotically standardized residuals</td>
</tr>
<tr>
<td>MAXASRES</td>
<td>ML asymptotically standardized residuals</td>
</tr>
<tr>
<td>WLSASRES</td>
<td>WLS asymptotically standardized residuals</td>
</tr>
<tr>
<td>DWLSASRS</td>
<td>DWLS asymptotically standardized residuals</td>
</tr>
<tr>
<td>UNROTATE</td>
<td>unrotated factor loadings</td>
</tr>
<tr>
<td>FCORR</td>
<td>matrix of factor correlations</td>
</tr>
<tr>
<td>UNIQUE_ V</td>
<td>unique variances</td>
</tr>
<tr>
<td>TRANSFORM</td>
<td>transformation matrix of rotation</td>
</tr>
<tr>
<td>LOADINGS</td>
<td>rotated factor loadings</td>
</tr>
<tr>
<td>STD_LOAD</td>
<td>standardized factor loadings</td>
</tr>
<tr>
<td>LSSCORE</td>
<td>factor regression scores for ULS method</td>
</tr>
<tr>
<td>SCORE</td>
<td>factor regression scores other than ULS method</td>
</tr>
</tbody>
</table>

The _NAME_ variable contains the name of the manifest variable corresponding to each row for the covariance, correlation, predicted, and residual matrices and con-
tains the name of the latent variable in case of factor regression scores. For other observations, _NAME_ is blank.

The unique variances and rotated loadings can be used as starting values in more difficult and constrained analyses.

If the model contains latent variables, the OUTSTAT= data set also contains the latent variable score regression coefficients and the predicted covariances between latent and manifest variables. You can use the latent variable score regression coefficients with PROC SCORE to compute factor scores.

If the analyzed matrix is a (corrected or uncorrected) covariance rather than a correlation matrix, the _TYPE_=STD or _TYPE_=USTD observation is not included in the OUTSTAT= data set. In this case, the standard deviations can be obtained from the diagonal elements of the covariance matrix. Dropping the _TYPE_=STD or _TYPE_=USTD observation prevents PROC SCORE from standardizing the observations before computing the factor scores.

**OUTWGT= SAS-data-set**

You can create an OUTWGT= data set that is of TYPE=WEIGHT and contains the weight matrix used in generalized, weighted, or diagonally weighted least-squares estimation. The inverse of the weight matrix is used in the corresponding fit function. The OUTWGT= data set contains the weight matrix on which the WRIDGE= and the WPENALTY= options are applied. For unweighted least-squares or maximum likelihood estimation, no OUTWGT= data set can be written. The last weight matrix used in maximum likelihood estimation is the predicted model matrix (observations with _TYPE_=MAXPRED) that is included in the OUTSTAT= data set.

For generalized and diagonally weighted least-squares estimation, the weight matrices \( W \) of the OUTWGT= data set contain all elements \( w_{ij} \), where the indices \( i \) and \( j \) correspond to all manifest variables used in the analysis. Let \( varnam_i \) be the name of the \( i \)th variable in the analysis. In this case, the OUTWGT= data set contains \( n \) observations with variables as displayed in the following table.

**Table 17.11. Contents of OUTWGT= data set for GLS and DWLS Estimation**

<table>
<thead>
<tr>
<th>Variable</th>
<th>Contents</th>
</tr>
</thead>
<tbody>
<tr>
<td><em>TYPE</em></td>
<td>WEIGHT (character)</td>
</tr>
<tr>
<td><em>NAME</em></td>
<td>name of variable ( varnam_i ) (character)</td>
</tr>
<tr>
<td>( varnam_1 )</td>
<td>weight ( w_{11} ) for variable ( varnam_1 ) (numeric)</td>
</tr>
<tr>
<td>( \vdots )</td>
<td>( \vdots )</td>
</tr>
<tr>
<td>( varnam_n )</td>
<td>weight ( w_{nn} ) for variable ( varnam_n ) (numeric)</td>
</tr>
</tbody>
</table>

For weighted least-squares estimation, the weight matrix \( W \) of the OUTWGT= data set contains only the nonredundant elements \( w_{ij,k} \). In this case, the OUTWGT= data set contains \( n(n + 1)(2n + 1)/6 \) observations with variables as follows.
Table 17.12. Contents of OUTWGT= data set for WLS Estimation

<table>
<thead>
<tr>
<th>Variable</th>
<th>Contents</th>
</tr>
</thead>
<tbody>
<tr>
<td><em>TYPE</em></td>
<td>WEIGHT (character)</td>
</tr>
<tr>
<td><em>NAME</em></td>
<td>name of variable varnam_i (character)</td>
</tr>
<tr>
<td><em>NAM2</em></td>
<td>name of variable varnam_j (character)</td>
</tr>
<tr>
<td><em>NAM3</em></td>
<td>name of variable varnam_k (character)</td>
</tr>
<tr>
<td>varnam_1</td>
<td>weight $w_{ij,k1}$ for variable varnam_1 (numeric)</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>varnam_n</td>
<td>weight $w_{ij,kn}$ for variable varnam_n (numeric)</td>
</tr>
</tbody>
</table>

Symmetric redundant elements are set to missing values.

Missing Values

If the DATA= data set contains raw data (rather than a covariance or correlation matrix), observations with missing values for any variables in the analysis are omitted from the computations. If a covariance or correlation matrix is read, missing values are allowed as long as every pair of variables has at least one nonmissing value.

Estimation Criteria

The following five estimation methods are available in PROC CALIS:

- unweighted least squares (ULS)
- generalized least squares (GLS)
- normal-theory maximum likelihood (ML)
- weighted least squares (WLS, ADF)
- diagonally weighted least squares (DWLS)

An INWGT= data set can be used to specify other than the default weight matrices $W$ for GLS, WLS, and DWLS estimation.

In each case, the parameter vector is estimated iteratively by a nonlinear optimization algorithm that optimizes a goodness-of-fit function $F$. When $n$ denotes the number of manifest variables, $S$ denotes the given sample covariance or correlation matrix for a sample with size $N$, and $C$ denotes the predicted moment matrix, then the fit function for unweighted least-squares estimation is

$$F_{ULS} = .5Tr[(S - C)^2]$$

For normal-theory generalized least-squares estimation, the function is

$$F_{GLS} = .5Tr[S^{-1}(S - C)^2]$$

For normal-theory maximum likelihood estimation, the function is

$$F_{ML} = Tr(SC^{-1}) - n + ln(det(C)) - ln(det(S))$$
The first three functions can be expressed by the generally weighted least-squares criterion (Browne 1982):

\[ F_{GW\,LS} = 0.5 \text{Tr}[(W^{-1}(S - C))^2] \]

For unweighted least squares, the weight matrix \( W \) is chosen as the identity matrix \( I \); for generalized least squares, the default weight matrix \( W \) is the sample covariance matrix \( S \); and for normal-theory maximum likelihood, \( W \) is the iteratively updated predicted moment matrix \( C \). The values of the normal-theory maximum likelihood function \( F_{ML} \) and the generally weighted least-squares criterion \( F_{GW\,LS} \) with \( W = C \) are asymptotically equivalent.

The goodness-of-fit function that is minimized in weighted least-squares estimation is

\[ F_{WL\,S} = \text{Vec}(s_{ij} - c_{ij}^T)W^{-1}\text{Vec}(s_{ij} - c_{ij}) \]

where \( \text{Vec}(s_{ij} - c_{ij}) \) denotes the vector of the \( n(n + 1)/2 \) elements of the lower triangle of the symmetric matrix \( S - C \), and \( W = (w_{ijkl}) \) is a positive definite symmetric matrix with \( n(n + 1)/2 \) rows and columns.

If the moment matrix \( S \) is considered as a covariance rather than a correlation matrix, the default setting of \( W = (w_{ijkl}) \) is the consistent but biased estimators of the asymptotic covariances \( c_{ijkl} \) of the sample covariance \( s_{ij} \) with the sample covariance \( s_{kl} \)

\[ w_{ijkl} = s_{ijkl} - s_{ij} s_{kl} \]

where

\[ s_{ijkl} = \frac{1}{N} \sum_{r=1}^{N} (z_{ri} - \bar{z}_i)(z_{rj} - \bar{z}_j)(z_{rk} - \bar{z}_k)(z_{rl} - \bar{z}_l) \]

The formula of the asymptotic covariances of uncorrected covariances (using the UCOV or NOINT option) is a straightforward generalization of this expression.

The resulting weight matrix \( W \) is at least positive semidefinite (except for rounding errors). Using the ASYCOV option, you can use Browne’s (1984, formula (3.8)) unbiased estimators

\[ w_{ijkl} = \frac{N(N - 1)}{(N - 2)(N - 3)}(s_{ijkl} - s_{ij} s_{kl}) - \frac{N}{(N - 2)(N - 3)}(s_{ik} s_{jl} + s_{il} s_{jk} - 2) \]

There is no guarantee that this weight matrix is positive semidefinite. However, the second part is of order \( O(N^{-1}) \) and does not destroy the positive semidefinite first
part for sufficiently large \(N\). For a large number of independent observations, default settings of the weight matrix \(W\) result in asymptotically distribution-free parameter estimates with unbiased standard errors and a correct \(\chi^2\) test statistic (Browne 1982, 1984).

If the moment matrix \(S\) is a correlation (rather than a covariance) matrix, the default setting of \(W = (w_{ij,kl})\) is the estimators of the asymptotic covariances \(\sigma_{ij,kl}\) of the correlations \(S = (s_{ij})\) (Browne and Shapiro 1986; DeLeeuw 1983)

\[
  w_{ij,kl} = r_{ij,kl} - \frac{1}{2} r_{ij} (r_{ii,kl} + r_{jj,kl}) - \frac{1}{2} r_{kl} (r_{kk,ij} + r_{ll,ij}) + \frac{1}{4} r_{ij} r_{kl} (r_{ii,kl} + r_{ii,kl} + r_{jj,kl} + r_{jj,kl})
\]

where

\[
  r_{ij,kl} = \frac{s_{ij,kl}}{s_{ii}s_{jj}s_{kk}s_{ll}}
\]

The asymptotic variances of the diagonal elements of a correlation matrix are 0. Therefore, the weight matrix computed by Browne and Shapiro’s formula is always singular. In this case the goodness-of-fit function for weighted least-squares estimation is modified to

\[
  F_{WLS} = \sum_{i=2}^{n} \sum_{j=1}^{n} \sum_{k=2}^{n} \sum_{l=1}^{n} w_{ij,kl} (s_{ij} - c_{ij})(s_{kl} - c_{kl}) + r \sum_{i=1}^{n} (s_{ii} - c_{ii})^2
\]

where \(r\) is the penalty weight specified by the WPENALTY=r option and the \(w_{ij,kl}\) are the elements of the inverse of the reduced \((n(n-1)/2) \times (n(n-1)/2)\) weight matrix that contains only the nonzero rows and columns of the full weight matrix \(W\). The second term is a penalty term to fit the diagonal elements of the moment matrix \(S\). The default value of \(r = 100\) can be decreased or increased by the WPENALTY= option. The often used value of \(r = 1\) seems to be too small in many cases to fit the diagonal elements of a correlation matrix properly. If your model does not fit the diagonal of the moment matrix \(S\), you can specify the NODIAG option to exclude the diagonal elements from the fit function.

Storing and inverting the huge weight matrix \(W\) in WLS estimation needs considerable computer resources. A compromise is found by implementing the DWLS method that uses only the diagonal of the weight matrix \(W\) from the WLS estimation in the minimization function

\[
  F_{DWLS} = Vec(s_{ij} - c_{ij})' diag(W)^{-1} Vec(s_{ij} - c_{ij})
\]

The statistical properties of DWLS estimates are still not known.

In generalized, weighted, or diagonally weighted least-squares estimation, you can change from the default settings of weight matrices \(W\) by using an INWGT= data set. Because the diagonal elements \(w_{ii,kl}\) of the weight matrix \(W\) are interpreted
as asymptotic variances of the sample covariances or correlations, they cannot be negative. The CALIS procedure requests a positive definite weight matrix that has positive diagonal elements.

### Relationships among Estimation Criteria

The five estimation functions, $F_{ULS}, F_{GLS}, F_{ML}, F_{WLS}, \text{and } F_{DWLS}$, belong to the following two groups:

- The functions $F_{ULS}, F_{GLS}, \text{and } F_{ML}$ take into account all $n^2$ elements of the symmetric residual matrix $S - C$. This means that the off-diagonal residuals contribute twice to $F$, as lower and as upper triangle elements.
- The functions $F_{WLS}$ and $F_{DWLS}$ take into account only the $n(n + 1)/2$ lower triangular elements of the symmetric residual matrix $S - C$. This means that the off-diagonal residuals contribute to $F$ only once.

The $F_{DWLS}$ function used in PROC CALIS differs from that used by the LISREL 7 program. Formula (1.25) of the LISREL 7 manual (Jöreskog and Sörbom 1988, p. 23) shows that LISREL groups the $F_{DWLS}$ function in the first group by taking into account all $n^2$ elements of the symmetric residual matrix $S - C$.

- Relationship between DWLS and WLS:
  - PROC CALIS: The $F_{DWLS}$ and $F_{WLS}$ estimation functions deliver the same results for the special case that the weight matrix $W$ used by WLS estimation is a diagonal matrix.
  - LISREL 7: This is not the case.

- Relationship between DWLS and ULS:
  - LISREL 7: The $F_{DWLS}$ and $F_{ULS}$ estimation functions deliver the same results for the special case that the diagonal weight matrix $W$ used by DWLS estimation is an identity matrix (contains only 1s).
  - PROC CALIS: To obtain the same results with $F_{DWLS}$ and $F_{ULS}$ estimation, set the diagonal weight matrix $W$ used in DWLS estimation to

$$w_{ii,kk} = \begin{cases} 
1 & \text{if } i = k \\
0.5 & \text{otherwise}
\end{cases}$$

Because the reciprocal elements of the weight matrix are used in the goodness-of-fit function, the off-diagonal residuals are weighted by a factor of 2.

### Testing Rank Deficiency in the Approximate Covariance Matrix

The inverse of the information matrix (or approximate Hessian matrix) is used for the covariance matrix of the parameter estimates, which is needed for the computation of approximate standard errors and modification indices. The numerical condition of the information matrix (computed as the crossproduct $J'J$ of the Jacobian matrix $J$) can be very poor in many practical applications, especially for the analysis of unscaled covariance data. The following four-step strategy is used for the inversion of the information matrix:
1. The inversion (usually of a normalized matrix $D^{-1}HD^{-1}$) is tried using a modified form of the Bunch and Kaufman (1977) algorithm, which allows the specification of a different singularity criterion for each pivot. The following three criteria for the detection of rank loss in the information matrix are used to specify thresholds:

- **ASING** specifies absolute singularity.
- **MSING** specifies relative singularity depending on the whole matrix norm.
- **VSING** specifies relative singularity depending on the column matrix norm.

If no rank loss is detected, the inverse of the information matrix is used for the covariance matrix of parameter estimates, and the next two steps are skipped.

2. The linear dependencies among the parameter subsets are displayed based on the singularity criteria.

3. If the number of parameters $t$ is smaller than the value specified by the G4= option (the default value is 60), the Moore-Penrose inverse is computed based on the eigenvalue decomposition of the information matrix. If you do not specify the NOPRINT option, the distribution of eigenvalues is displayed, and those eigenvalues that are set to zero in the Moore-Penrose inverse are indicated. You should inspect this eigenvalue distribution carefully.

4. If PROC CALIS did not set the right subset of eigenvalues to zero, you can specify the COVSING= option to set a larger or smaller subset of eigenvalues to zero in a further run of PROC CALIS.

---

**Approximate Standard Errors**

Except for unweighted and diagonally weighted least-squares estimation, approximate standard errors can be computed as the diagonal elements of the matrix

$$\frac{c}{NM} \mathbf{H}^{-1}, \quad \text{where}$$

$$NM = \begin{cases} (N - 1) & \text{if the CORR or COV matrix is analyzed} \\ N & \text{if the UCORR or UCOV matrix is analyzed and the intercept variable is not used in the model} \end{cases}$$

The matrix $\mathbf{H}$ is the approximate Hessian matrix of $F$ evaluated at the final estimates, $c = 1$ for the WLS estimation method, $c = 2$ for the GLS and ML method, and $N$ is the sample size. If a given correlation or covariance matrix is singular, PROC CALIS offers two ways to compute a generalized inverse of the information matrix and, therefore, two ways to compute approximate standard errors of implicitly constrained parameter estimates, $t$ values, and modification indices. Depending on the G4= specification, either a Moore-Penrose inverse or a G2 inverse is computed. The expensive Moore-Penrose inverse computes an estimate of the null space using
an eigenvalue decomposition. The cheaper G2 inverse is produced by sweeping the
linearly independent rows and columns and zeroing out the dependent ones. The
information matrix, the approximate covariance matrix of the parameter estimates,
and the approximate standard errors are not computed in the cases of unweighted or
diagonally weighted least-squares estimation.

Assessment of Fit

This section contains a collection of formulas used in computing indices to assess the
goodness of fit by PROC CALIS. The following notation is used:

- \( N \) for the sample size
- \( n \) for the number of manifest variables
- \( t \) for the number of parameters to estimate
- \( \text{NM} = \begin{cases} 
(N - 1) & \text{if the CORR or COV matrix is analyzed} \\
& \text{or the intercept variable is not used in the model} \\
N & \text{if the UCORR or UCOV matrix is analyzed} \\
& \text{and the intercept variable is not used in the model}
\end{cases} \)
- \( df \) for the degrees of freedom
- \( \gamma = \mathbf{X} \) for the \( t \) vector of optimal parameter estimates
- \( S = (s_{ij}) \) for the \( n \times n \) input COV, CORR, UCOV, or UCORR matrix
- \( C = (c_{ij}) = \Sigma = \Sigma(\gamma) \) for the predicted model matrix
- \( W \) for the weight matrix (\( W = I \) for ULS, \( W = S \) for default GLS, and \( W = C \) for ML estimates)
- \( U \) for the \( n^2 \times n^2 \) asymptotic covariance matrix of sample covariances
- \( \Phi(x|\lambda, df) \) for the cumulative distribution function of the noncentral chi-
squared distribution with noncentrality parameter \( \lambda \)

The following notation is for indices that allow testing hierarchical models by a \( \chi^2 \)
difference test:

- \( f_0 \) for the function value of a baseline or null model
- \( df_0 \) for the degrees of freedom of a baseline or null model
- \( f_{\text{min}} = F \) for the function value of the fitted model
- \( df_{\text{min}} = df \) for the degrees of freedom of the fitted model

Residuals

PROC CALIS computes four types of residuals and writes them to the OUTSTAT=
data set:

- Simple Unnormed Residuals

\[
Res = S - C, \quad Res_{ij} = s_{ij} - c_{ij}
\]
The unnormed residuals are displayed whenever the PALL, the PRINT, or the RESIDUAL option is specified.

- **Variance Standardized Residuals**

\[
V S R e s_{ij} = \frac{s_{ij} - c_{ij}}{\sqrt{s_{ii}s_{jj}}}
\]

The variance standardized residuals are displayed when you specify

- the PALL, the PRINT, or the RESIDUAL option and METHOD=NONE, METHOD=ULS, or METHOD=DWLS
- RESIDUAL=VARSTAND

The variance standardized residuals are equal to those computed by the EQS 3 program (Bentler 1989).

- **Asymptotically Standardized Residuals**

\[
A S R e s_{ij} = \frac{s_{ij} - c_{ij}}{\sqrt{c_{ij,ij}}}, \quad \text{where}
\]

\[
c_{ij,ij} = diag(U - J Cov(\gamma)J^\prime)_{ij}
\]

The matrix J is the \(n^2 \times t\) Jacobian matrix \(d\Sigma / d\gamma\), and \(Cov(\gamma)\) is the \(t \times t\) asymptotic covariance matrix of parameter estimates (the inverse of the information matrix). Asymptotically standardized residuals are displayed when one of the following conditions is met:

- The PALL, the PRINT, or the RESIDUAL option is specified, and METHOD=ML, METHOD=GLS, or METHOD=WLS, and the expensive information and Jacobian matrices are computed for some other reason.
- RESIDUAL=ASYSTAND is specified.

The asymptotically standardized residuals are equal to those computed by the LISREL 7 program (Jöreskog and Sörbom 1988) except for the denominator \(N M\) in the definition of matrix \(U\).

- **Normalized Residuals**

\[
N R e s_{ij} = \frac{s_{ij} - c_{ij}}{u_{ij,ij}^{1/2}}
\]

where the diagonal elements \(u_{ij,ij}\) of the \(n^2 \times n^2\) asymptotic covariance matrix \(U\) of sample covariances are defined for method

- GLS as \(u_{ij,ij} = \frac{1}{N M}(s_{ii}s_{jj} + s_{ij}^2)\)
- ML as \(u_{ij,ij} = \frac{1}{N M}(c_{ii}c_{jj} + c_{ij}^2)\)
- WLS as \(u_{ij,ij} = w_{ij,ij}\)
Normalized residuals are displayed when one of the following conditions is met:

- The PALL, the PRINT, or the RESIDUAL option is specified, and METHOD=ML, METHOD=GLS, or METHOD=WLS, and the expensive information and Jacobian matrices are not computed for some other reason.
- RESIDUAL=NORM is specified.

The normalized residuals are equal to those computed by the LISREL VI program (Jöreskog and Sörbom 1985) except for the definition of the denominator $N_M$ in matrix $U$.

For estimation methods that are not BGLS estimation methods (Browne 1982, 1984), such as METHOD=NONE, METHOD=ULS, or METHOD=DWLS, the assumption of an asymptotic covariance matrix $U$ of sample covariances does not seem to be appropriate. In this case, the normalized residuals should be replaced by the more relaxed variance standardized residuals. Computation of asymptotically standardized residuals requires computing the Jacobian and information matrices. This is computationally very expensive and is done only if the Jacobian matrix has to be computed for some other reason, that is, if at least one of the following items is true:

- The default, PRINT, or PALL displayed output is requested, and neither the NOMOD nor NOSTDERR option is specified.
- Either the MODIFICATION (included in PALL), PCOVES, or STDERR (included in default, PRINT, and PALL output) option is requested or RESIDUAL=ASYSTAND is specified.
- The LEVMAR or NEWRAP optimization technique is used.
- An OUTRAM= data set is specified without using the NOSTDERR option.
- An OUTEST= data set is specified without using the NOSTDERR option.

Since normalized residuals use an overestimate of the asymptotic covariance matrix of residuals (the diagonal of $U$), the normalized residuals cannot be larger than the asymptotically standardized residuals (which use the diagonal of $U - J\text{Cov}(\gamma)J'$).

Together with the residual matrices, the values of the average residual, the average off-diagonal residual, and the rank order of the largest values are displayed. The distribution of the normalized and standardized residuals is displayed also.

**Goodness-of-Fit Indices**

The following items are computed for all five kinds of estimation: ULS, GLS, ML, WLS, and DWLS. All these indices are written to the OUTRAM= data set. The goodness of fit (GFI), adjusted goodness of fit (AGFI), and root mean square residual (RMR) are computed as in the LISREL VI program of Jöreskog and Sörbom (1985).

- **Goodness-of-Fit Index**
  
  The goodness-of-fit index for the ULS, GLS, and ML estimation methods is

  $$GFI = 1 - \frac{Tr((W^{-1}(S - C))^2)}{Tr((W^{-1}S)^2)}$$
but for WLS and DWLS estimation, it is

\[ GFI = 1 - \frac{Vec(s_{ij} - c_{ij})'W^{-1}Vec(s_{ij} - c_{ij})}{Vec(s_{ij})'W^{-1}Vec(s_{ij})} \]

where \( W = diag \) for DWLS estimation, and \( Vec(s_{ij} - c_{ij}) \) denotes the vector of the \( n(n+1)/2 \) elements of the lower triangle of the symmetric matrix \( S-C \). For a constant weight matrix \( W \), the goodness-of-fit index is 1 minus the ratio of the minimum function value and the function value before any model has been fitted. The GFI should be between 0 and 1. The data probably do not fit the model if the GFI is negative or much larger than 1.

- **Adjusted Goodness-of-Fit Index**
  The AGFI is the GFI adjusted for the degrees of freedom of the model

\[ AGFI = 1 - \frac{n(n+1)}{2df}(1 - GFI) \]

The AGFI corresponds to the GFI in replacing the total sum of squares by the mean sum of squares.

**CAUTION:**
- Large \( n \) and small \( df \) can result in a negative AGFI. For example, GFI=0.90, \( n=19 \), and \( df=2 \) result in an AGFI of -8.5.
- AGFI is not defined for a saturated model, due to division by \( df = 0 \).
- AGFI is not sensitive to losses in \( df \).

The AGFI should be between 0 and 1. The data probably do not fit the model if the AGFI is negative or much larger than 1. For more information, refer to Mulaik et al. (1989).

- **Root Mean Square Residual**
  The RMR is the mean of the squared residuals:

\[ RMR = \sqrt{\frac{2}{n(n+1)} \sum_{i} \sum_{j} (s_{ij} - c_{ij})^2} \]

- **Parsimonious Goodness-of-Fit Index**
  The PGFI (Mulaik et al. 1989) is a modification of the GFI that takes the parsimony of the model into account:

\[ PGFI = \frac{df_{min}}{df_0} GFI \]

The PGFI uses the same parsimonious factor as the parsimonious normed Bentler-Bonett index (James, Mulaik, and Brett 1982).
**$\chi^2$ Indices**

The following items are transformations of the overall $\chi^2$ value and in general depend on the sample size $N$. Some indices, for example, the Bentler and Bonett indices, allow you to test hierarchical models by a $\chi^2$ difference. These indices are not computed for ULS or DWLS estimates.

- **Uncorrected $\chi^2$**
  The overall $\chi^2$ measure is the optimum function value $F$ multiplied by $N - 1$ if a CORR or COV matrix is analyzed, or multiplied by $N$ if a UCORR or UCOV matrix is analyzed. This gives the likelihood ratio test statistic for the null hypothesis that the predicted matrix $C$ has the specified model structure against the alternative that $C$ is unconstrained. The $\chi^2$ test is valid only if the observations are independent and identically distributed, the analysis is based on the nonstandardized sample covariance matrix $S$, and the sample size $N$ is sufficiently large (Browne 1982; Bollen 1989b; Jöreskog and Sörbom 1985). For ML and GLS estimates, the variables must also have an approximately multivariate normal distribution. The notation Prob>Chi**2 means “the probability under the null hypothesis of obtaining a greater $\chi^2$ statistic than that observed.”

\[
\chi^2 = N M \times F
\]

where $F$ is the function value at the minimum.

- **$\chi^2_0$ Value of the Baseline Model**
  The $\chi^2_0$ value of the baseline model

\[
\chi^2_0 = N M \times f_0
\]

and the corresponding degrees of freedom $df_0$ can be used (in large samples) to evaluate the gain of explanation by fitting the specific model (Bentler 1989).

- **RMSEA Index (Steiger and Lind 1980)**
  The Steiger and Lind (1980) root mean squared error approximation (RMSEA) coefficient is

\[
\epsilon_\alpha = \sqrt{\max\left(\frac{F}{df} - \frac{1}{NM}, 0\right)}
\]

The lower and upper limits of the confidence interval are computed using the cumulative distribution function of the noncentral chi-squared distribution $\Phi(x|\lambda, df) = \alpha$, with $x = NM \times F$, $\lambda_L$ satisfying $\Phi(x|\lambda_L, df) = 1 - \frac{\alpha}{2}$, and $\lambda_U$ satisfying $\Phi(x|\lambda_U, df) = \frac{\alpha}{2}$:

\[
(\epsilon_{\alpha_L}; \epsilon_{\alpha_U}) = \left(\sqrt{\frac{\lambda_L}{NM \times df}}; \sqrt{\frac{\lambda_U}{NM \times df}}\right)
\]

Refer to Browne and Du Toit (1992) for more details. The size of the confidence interval is defined by the option ALPHARMS=$\alpha$, $0 \leq \alpha \leq 1$. The default is $\alpha = 0.1$, which corresponds to the 90% confidence interval for the RMSEA.
Chapter 17. The CALIS Procedure

- **Probability for Test of Close Fit (Browne and Cudeck 1993)**
  The traditional exact $\chi^2$ test hypothesis $H_0: \epsilon_\alpha = 0$ is replaced by the null hypothesis of close fit $H_0: \epsilon_\alpha \leq 0.05$ and the exceedance probability $P$ is computed as
  \[ P = 1 - \Phi(x|\lambda^*, df) \]
  where $x = NM * F$ and $\lambda^* = 0.05^2 * NM * df$. The null hypothesis of close fit is rejected if $P$ is smaller than a prespecified level (for example, $P < 0.05$).

- **Expected Cross Validation Index (Browne and Cudeck 1993)**
  For GLS and WLS, the estimator $c$ of the ECVI is linearly related to AIC:
  \[ c = F(S, C) + \frac{2t}{NM} \]
  For ML estimation, $c_{ML}$ is used.
  \[ c_{ML} = F_{ML}(S, C) + \frac{2t}{NM - n - 1} \]
  The confidence interval $(c_L; c_U)$ for $c$ is computed using the cumulative distribution function $\Phi(x|\lambda, df)$ of the noncentral chi-squared distribution,
  \[ (c_L; c_U) = \left( \frac{\lambda_L + nnt}{NM}; \frac{\lambda_U + nnt}{NM} \right) \]
  with $nnt = n(n + 1)/2 + t$, $x = NM * F$, $\Phi(x|\lambda_U, df) = 1 - \frac{\alpha}{2}$, and $\Phi(x|\lambda_L, df) = \frac{\alpha}{2}$. The confidence interval $(c^*_L; c^*_U)$ for $c_{ML}$ is
  \[ (c^*_L; c^*_U) = \left( \frac{\lambda^*_L + nnt}{NM - n - 1}; \frac{\lambda^*_U + nnt}{NM - n - 1} \right) \]
  where $nnt = n(n + 1)/2 + t$, $x = (NM - n - 1) * F$, $\Phi(x|\lambda^*_L, df) = 1 - \frac{\alpha}{2}$ and $\Phi(x|\lambda^*_U, df) = \frac{\alpha}{2}$. Refer to Browne and Cudeck (1993). The size of the confidence interval is defined by the option ALPHAECV=$\alpha$, $0 \leq \alpha \leq 1$. The default is $\alpha = 0.1$, which corresponds to the 90% confidence interval $[.05, .95]$.

- **Comparative Fit Index (Bentler 1989)**
  \[ CFI = 1 - \frac{\max(NM * f_{min} - df_{min}, 0)}{\max(NM * f_0 - df_0, 0)} \]

- **Adjusted $\chi^2$ Value (Browne 1982)**
  If the variables are $n$-variate elliptic rather than normal and have significant amounts of multivariate kurtosis (leptokurtic or platykurtic), the $\chi^2$ value can be adjusted to
  \[ \chi^2_{ell} = \frac{\chi^2}{\eta_2} \]
  where $\eta_2$ is the multivariate relative kurtosis coefficient.
• **Normal Theory Reweighted LS $\chi^2$ Value**
This index is displayed only if METHOD=ML. Instead of the function value $F_{ML}$, the reweighted goodness-of-fit function $F_{GWLS}$ is used,

$$\chi^2_{GWLS} = NM \cdot F_{GWLS}$$

where $F_{GWLS}$ is the value of the function at the minimum.

• **Akaike's Information Criterion (AIC) (Akaike 1974; Akaike 1987)**
This is a general criterion for estimating the best number of parameters to include in a model. The number of parameters that yields the smallest value of AIC is considered best.

$$AIC = \chi^2 - 2df$$

• **Consistent Akaike's Information Criterion (CAIC) (Bozdogan 1987)**
This is another criterion, similar to AIC, for determining the best number of parameters. The number of parameters that yields the smallest value of CAIC is considered best. CAIC is preferred by some people to AIC or the $\chi^2$ test.

$$CAIC = \chi^2 - (\ln(N) + 1)df$$

• **Schwarz’s Bayesian Criterion (SBC) (Schwarz 1978; Sclove 1987)**
This is another criterion, similar to AIC, for determining the best number of parameters. The number of parameters that yields the smallest value of SBC is considered best. SBC is preferred by some people to AIC or the $\chi^2$ test.

$$SBC = \chi^2 - \ln(N)df$$

• **McDonald’s Measure of Centrality (McDonald and Hartmann 1992)**

$$CENT = \exp\left(-\frac{(\chi^2 - df)}{2N}\right)$$

• **Parsimonious Normed Fit Index (James, Mulaik, and Brett 1982)**
The PNFI is a modification of Bentler-Bonett’s normed fit index that takes parsimony of the model into account,

$$PNFI = \frac{df_{\text{min}}}{df_0} \left(\frac{f_0 - f_{\text{min}}}{f_0}\right)$$

The PNFI uses the same parsimonious factor as the parsimonious GFI of Mulaik et al. (1989).

• **Z-Test (Wilson and Hilferty 1931)**
The Z-Test of Wilson and Hilferty assumes an $n$- variate normal distribution:

$$Z = \sqrt{\frac{\chi^2}{df}} - \frac{1 - \frac{2}{\sqrt{2df}}}{\sqrt{\frac{2}{2df}}}$$

Refer to McArdle (1988) and Bishop, Fienberg, and Holland (1977, p. 527) for an application of the Z-Test.
• **Nonnormed Coefficient (Bentler and Bonett 1980)**

\[
\rho = \frac{f_0/df_0 - f_{\min}/df_{\min}}{f_0/df_0 - 1/NNM}
\]

Refer to Tucker and Lewis (1973).

• **Normed Coefficient (Bentler and Bonett 1980)**

\[
\Delta = \frac{f_0 - f_{\min}}{f_0}
\]

Mulaik et al. (1989) recommend the parsimonious weighted form PNFI.

• **Normed Index \( \rho_1 \) (Bollen 1986)**

\[
\rho_1 = \frac{f_0/df_0 - f_{\min}/df_{\min}}{f_0/df_0}
\]

\( \rho_1 \) is always less than or equal to 1; \( \rho_1 < 0 \) is unlikely in practice. Refer to the discussion in Bollen (1989a).

• **Nonnormed Index \( \Delta_2 \) (Bollen 1989a)**

\[
\Delta_2 = \frac{f_0 - f_{\min}}{f_0 - df_{NNM}}
\]

is a modification of Bentler & Bonett’s \( \Delta \) that uses \( df \) and “lessens the dependence” on \( N \). Refer to the discussion in Bollen (1989b). \( \Delta_2 \) is identical to Mulaik et al.’s (1989) IFI2 index.

• **Critical N Index (Hoelter 1983)**

\[
CN = \frac{\chi^2_{\text{crit}}}{F} + 1
\]

where \( \chi^2_{\text{crit}} \) is the critical chi-square value for the given \( df \) degrees of freedom and probability \( \alpha = 0.05 \), and \( F \) is the value of the estimation criterion (minimization function). Refer to Bollen (1989b, p. 277). Hoelter (1983) suggests that CN should be at least 200; however, Bollen (1989b) notes that the CN value may lead to an overly pessimistic assessment of fit for small samples.
Squared Multiple Correlation

The following are measures of the squared multiple correlation for manifest and endogenous variables and are computed for all five estimation methods: ULS, GLS, ML, WLS, and DWLS. These coefficients are computed as in the LISREL VI program of Jöreskog and Sörbom (1985). The DETAE, DETSE, and DETMV determination coefficients are intended to be global means of the squared multiple correlations for different subsets of model equations and variables. These coefficients are displayed only when you specify the PDETERM option with a RAM or LINEQS model.

- $R^2$ Values Corresponding to Endogenous Variables

$$R^2_i = 1 - \frac{\text{var}(\zeta_i)}{\text{var}(\eta_i)}$$

- Total Determination of All Equations

$$DETAE = 1 - \frac{\text{det}(\Theta, \Psi)}{\text{det}(\text{Cov}(y,x,\eta))}$$

- Total Determination of the Structural Equations

$$DETSE = 1 - \frac{\text{det}(\hat{\Psi})}{\text{det}(\text{Cov}(\eta))}$$

- Total Determination of the Manifest Variables

$$DET MV = 1 - \frac{\text{det}(\hat{\Theta})}{\text{det}(S)}$$

CAUTION: In the LISREL program, the structural equations are defined by specifying the BETA matrix. In PROC CALIS, a structural equation has a dependent left-hand-side variable that appears at least once on the right-hand side of another equation, or the equation has at least one right-hand-side variable that is the left-hand-side variable of another equation. Therefore, PROC CALIS sometimes identifies more equations as structural equations than the LISREL program does.

Measures of Multivariate Kurtosis

In many applications, the manifest variables are not even approximately multivariate normal. If this happens to be the case with your data set, the default generalized least-squares and maximum likelihood estimation methods are not appropriate, and
you should compute the parameter estimates and their standard errors by an asymptotically distribution-free method, such as the WLS estimation method. If your manifest variables are multivariate normal, then they have a zero relative multivariate kurtosis, and all marginal distributions have zero kurtosis (Browne 1982). If your DATA= data set contains raw data, PROC CALIS computes univariate skewness and kurtosis and a set of multivariate kurtosis values. By default, the values of univariate skewness and kurtosis are corrected for bias (as in PROC UNIVARIATE), but using the BIASKUR option enables you to compute the uncorrected values also. The values are displayed when you specify the PROC CALIS statement option KURTOSIS.

- **Corrected Variance for Variable** \( z_j \)

\[
\sigma_j^2 = \frac{1}{N - 1} \sum_{i}^{N} (z_{ij} - \bar{z}_j)^2
\]

- **Corrected Univariate Skewness for Variable** \( z_j \)

\[
\gamma_{1(j)} = \frac{N}{(N - 1)(N - 2)} \frac{\sum_{i}^{N} (z_{ij} - \bar{z}_j)^3}{\sigma_j^3}
\]

- **Uncorrected Univariate Skewness for Variable** \( z_j \)

\[
\gamma_{1(j)} = \frac{N \sum_{i}^{N} (z_{ij} - \bar{z}_j)^3}{\sqrt{N[\sum_{i}^{N} (z_{ij} - \bar{z}_j)^2]^3}}
\]

- **Corrected Univariate Kurtosis for Variable** \( z_j \)

\[
\gamma_{2(j)} = \frac{N(N + 1)}{(N - 1)(N - 2)(N - 3)} \frac{\sum_{i}^{N} (z_{ij} - \bar{z}_j)^4}{\sigma_j^4} - \frac{3(N - 1)^2}{(N - 2)(N - 3)}
\]

- **Uncorrected Univariate Kurtosis for Variable** \( z_j \)

\[
\gamma_{2(j)} = \frac{N \sum_{i}^{N} (z_{ij} - \bar{z}_j)^4}{[\sum_{i}^{N} (z_{ij} - \bar{z}_j)^2]^2} - 3
\]

- **Mardia’s Multivariate Kurtosis**

\[
\gamma_2 = \frac{1}{N} \sum_{i}^{N} [(z_i - \bar{z})' S^{-1} (z_i - \bar{z})]^2 - n(n + 2)
\]
• Relative Multivariate Kurtosis

\[ \eta_2 = \frac{\gamma_2 + n(n + 2)}{n(n + 2)} \]

• Normalized Multivariate Kurtosis

\[ \kappa_0 = \frac{\gamma_2}{\sqrt{8n(n + 2)/N}} \]

• Mardia Based Kappa

\[ \kappa_1 = \frac{\gamma_2}{n(n + 2)} \]

• Mean Scaled Univariate Kurtosis

\[ \kappa_2 = \frac{1}{3n} \sum_{j}^{n} \gamma_{2(j)} \]

• Adjusted Mean Scaled Univariate Kurtosis

\[ \kappa_3 = \frac{1}{3n} \sum_{j}^{n} \gamma_{2(j)}^* \]

with

\[ \gamma_{2(j)}^* = \begin{cases} \gamma_{2(j)}, & \text{if } \gamma_{2(j)} > \frac{-6}{n+2} \\ \frac{-6}{n+2}, & \text{otherwise} \end{cases} \]

If variable \( Z_j \) is normally distributed, the uncorrected univariate kurtosis \( \gamma_{2(j)} \) is equal to 0. If \( Z \) has an \( n \)-variate normal distribution, Mardia’s multivariate kurtosis \( \gamma_2 \) is equal to 0. A variable \( Z_j \) is called leptokurtic if it has a positive value of \( \gamma_{2(j)} \) and is called platykurtic if it has a negative value of \( \gamma_{2(j)} \). The values of \( \kappa_1, \kappa_2, \) and \( \kappa_3 \) should not be smaller than a lower bound (Bentler 1985):

\[ \kappa \geq \frac{-2}{n + 2} \]

PROC CALIS displays a message if this happens.

If weighted least-squares estimates (METHOD=WLS or METHOD=ADF) are specified and the weight matrix is computed from an input raw data set, the CALIS procedure computes two further measures of multivariate kurtosis:
• **Multivariate Mean Kappa**

\[ \kappa_A = \frac{1}{m} \sum_{i}^{n} \sum_{j}^{1} \sum_{k}^{n} \sum_{l}^{1} \tilde{\kappa}_{ij,kl} - 1 \]

where

\[ \tilde{\kappa}_{ij,kl} = \frac{s_{ij,kl}}{s_{ij} s_{kl} + s_{kl} s_{jl} + s_{jl} s_{jk}} \]

and \( m = n(n + 1)(n + 2)(n + 3)/24 \) is the number of elements in the vector \( s_{ij,kl} \) (Bentler 1985).

• **Multivariate Least-Squares Kappa**

\[ \kappa_5 = \frac{s'_4 s_2}{s_2 s_2} - 1 \]

where

\[ s_{ij,kl} = \frac{1}{N} \sum_{r=1}^{N} (z_{ri} - \bar{z}_i)(z_{rj} - \bar{z}_j)(z_{rk} - \bar{z}_k)(z_{rl} - \bar{z}_l) \]

\( s_4 \) is the vector of the \( s_{ij,kl} \), and \( s_2 \) is the vector of the elements in the denominator of \( \kappa \) (Bentler 1985).

The occurrence of significant nonzero values of Mardia’s multivariate kurtosis \( \gamma_2 \) and significant amounts of some of the univariate kurtosis values \( \gamma_{2(j)} \) indicate that your variables are not multivariate normal distributed. Violating the multivariate normality assumption in (default) generalized least-squares and maximum likelihood estimation usually leads to the wrong approximate standard errors and incorrect fit statistics based on the \( \chi^2 \) value. In general, the parameter estimates are more stable against violation of the normal distribution assumption. For more details, refer to Browne (1974, 1982, 1984).

### Initial Estimates

Each optimization technique requires a set of initial values for the parameters. To avoid local optima, the initial values should be as close as possible to the globally optimal solution. You can check for local optima by running the analysis with several different sets of initial values; the RANDOM= option in the PROC CALIS statement is useful in this regard.

• **RAM and LINEQS**: There are several default estimation methods available in PROC CALIS for initial values of parameters in a linear structural equation model specified by a RAM or LINEQS model statement, depending on the form of the specified model:
Automatic Variable Selection

- two-stage least-squares estimation
- instrumental variable method (Hägglund 1982; Jennrich 1987)
- approximative factor analysis method
- ordinary least-squares estimation
- estimation method of McDonald (McDonald and Hartmann 1992)

- FACTOR: For default (exploratory) factor analysis, PROC CALIS computes initial estimates for factor loadings and unique variances by an algebraic method of approximate factor analysis. If you use a MATRIX statement together with a FACTOR model specification, initial values are computed by McDonald’s (McDonald and Hartmann 1992) method if possible. McDonald’s method of computing initial values works better if you scale the factors by setting the factor correlations to 1 rather than setting the loadings of the reference variables equal to 1. If none of the two methods seems to be appropriate, the initial values are set by the START= option.

- COSAN: For the more general COSAN model, there is no default estimation method for the initial values. In this case, the START= or RANDOM= option can be used to set otherwise unassigned initial values.

Poor initial values can cause convergence problems, especially with maximum likelihood estimation. You should not specify a constant initial value for all parameters since this would produce a singular predicted model matrix in the first iteration. Sufficiently large positive diagonal elements in the central matrices of each model matrix term provide a nonnegative definite initial predicted model matrix. If maximum likelihood estimation fails to converge, it may help to use METHOD=LSML, which uses the final estimates from an unweighted least-squares analysis as initial estimates for maximum likelihood. Or you can fit a slightly different but better-behaved model and produce an OUTRAM= data set, which can then be modified in accordance with the original model and used as an INRAM= data set to provide initial values for another analysis.

If you are analyzing a covariance or scalar product matrix, be sure to take into account the scales of the variables. The default initial values may be inappropriate when some variables have extremely large or small variances.

Automatic Variable Selection

You can use the VAR statement to reorder the variables in the model and to delete the variables not used. Using the VAR statement saves memory and computation time. If a linear structural equation model using the RAM or LINEQS statement (or an INRAM= data set specifying a RAM or LINEQS model) does not use all the manifest variables given in the input DATA= data set, PROC CALIS automatically deletes those manifest variables not used in the model.

In some special circumstances, the automatic variable selection performed for the RAM and LINEQS statements may be inappropriate, for example, if you are interested in modification indices connected to some of the variables that are not used in
the model. You can include such manifest variables as exogenous variables in the analysis by specifying constant zero coefficients.

For example, the first three steps in a stepwise regression analysis of the Werner Blood Chemistry data (Jöreskog and Sörbom 1988, p. 111) can be performed as follows:

```
PROC CALIS DATA=DIXON METHOD=GLS NOBS=180 PRINT MOD;
LINEQS Y=0 X1+0 X2+0 X3+0 X4+0 X5+0 X6+0 X7+E;
STD E=VAR;
RUN;
PROC CALIS DATA=DIXON METHOD=GLS NOBS=180 PRINT MOD;
LINEQS Y=G1 X1+0 X2+0 X3+0 X4+0 X5+0 X6+0 X7+E;
STD E=VAR;
RUN;
PROC CALIS DATA=DIXON METHOD=GLS NOBS=180 PRINT MOD;
LINEQS Y=G1 X1+0 X2+0 X3+0 X4+0 X5+G6 X6+0 X7+E;
STD E=VAR;
RUN;
```

Using the COSAN statement does not automatically delete those variables from the analysis that are not used in the model. You can use the output of the predetermined values in the predicted model matrix (PREDET option) to detect unused variables. Variables that are not used in the model are indicated by 0 rows and columns in the predetermined predicted model matrix.

### Exogenous Manifest Variables

If there are exogenous manifest variables in the linear structural equation model, then there is a one-to-one relationship between the given covariances and corresponding estimates in the central model matrix (\( \mathbf{P} \) or \( \Phi \)). In general, using exogenous manifest variables reduces the degrees of freedom since the corresponding sample correlations or covariances are not part of the exogenous information provided for the parameter estimation. See the section “Counting the Degrees of Freedom” on page 522 for more information.

If you specify a RAM or LINEQS model statement, or if such a model is recognized in an INRAM= data set, those elements in the central model matrices that correspond to the exogenous manifest variables are reset to the sample values after computing covariances or correlations within the current BY group.

The COSAN statement does not automatically set the covariances in the central model matrices that correspond to manifest exogenous variables.

You can use the output of the predetermined values in the predicted model matrix (PREDET option) that correspond to manifest exogenous variables to see which of the manifest variables are exogenous variables and to help you set the corresponding locations of the central model matrices with their covariances.

The following two examples show how different the results of PROC CALIS can be if manifest variables are considered either as endogenous or as exogenous variables.
In both examples, a correlation matrix $S$ is tested against an identity model matrix $C$; that is, no parameter is estimated. The three runs of the first example (specified by the COSAN, LINEQS, and RAM statements) consider the two variables $y$ and $x$ as endogenous variables.

```sas
TITLE2 'Data: FULLER (1987, p.18)';
DATA CORN;
   INPUT Y X;
   DATALINES;
     86 70
   115 97
   90 53
   86 64
   110 95
   91 64
   99 50
   96 70
   99 94
   104 69
   96 51;

TITLE3 'Endogenous Y and X';
PROC CALIS DATA=CORN;
   COSAN CORR(2,IDE);
RUN;
PROC CALIS DATA=CORN;
   LINEQS
      Y=EY,
      X=EX;
   STD  EY EX=2 * 1;
RUN;
PROC CALIS DATA=CORN;
   RAM
     1 1 3 1.,
     1 2 4 1.,
     2 3 3 1.,
     2 4 4 1.;
RUN;

The two runs of the second example (specified by the LINEQS and RAM statements) consider $y$ and $x$ as exogenous variables.

```sas
TITLE3 'Exogenous Y and X';
PROC CALIS DATA=CORN;
   STD  Y X=2 * 1;
RUN;
PROC CALIS DATA=CORN;
   RAM
     2 1 1 1.,
     2 2 2 1.;
RUN;
```
Exogenous $x, y$

Endogenous $x, y$

**Figure 17.5.** Exogenous and Endogenous Variables

The LINEQS and the RAM model statements set the covariances (correlations) of exogenous manifest variables in the estimated model matrix and automatically reduce the degrees of freedom.

### Use of Optimization Techniques

No algorithm for optimizing general nonlinear functions exists that will always find the global optimum for a general nonlinear minimization problem in a reasonable amount of time. Since no single optimization technique is invariably superior to others, PROC CALIS provides a variety of optimization techniques that work well in various circumstances. However, you can devise problems for which none of the techniques in PROC CALIS will find the correct solution. All optimization techniques in PROC CALIS use $O(n^2)$ memory except the conjugate gradient methods, which use only $O(n)$ of memory and are designed to optimize problems with many parameters.

The PROC CALIS statement NLOPTIONS can be especially helpful for tuning applications with nonlinear equality and inequality constraints on the parameter estimates. Some of the options available in NLOPTIONS may also be invoked as PROC CALIS options. The NLOPTIOns statement can specify almost the same options as the SAS/OR NLP procedure.

Nonlinear optimization requires the repeated computation of

- the function value (optimization criterion)
- the gradient vector (first-order partial derivatives)
- for some techniques, the (approximate) Hessian matrix (second-order partial derivatives)
- values of linear and nonlinear constraints
- the first-order partial derivatives (Jacobian) of nonlinear constraints

For the criteria used by PROC CALIS, computing the gradient takes more computer time than computing the function value, and computing the Hessian takes much more computer time and memory than computing the gradient, especially when there are many parameters to estimate. Unfortunately, optimization techniques that do not use the Hessian usually require many more iterations than techniques that do use the (approximate) Hessian, and so they are often slower. Techniques that do not use the
Hessian also tend to be less reliable (for example, they may terminate at local rather than global optima).

The available optimization techniques are displayed in Table 17.13 and can be chosen by the TECH= \textit{name} option.

\textbf{Table 17.13.} Optimization Techniques

<table>
<thead>
<tr>
<th>TECH=</th>
<th>Optimization Technique</th>
</tr>
</thead>
<tbody>
<tr>
<td>LEVMAR</td>
<td>Levenberg-Marquardt Method</td>
</tr>
<tr>
<td>TRUREG</td>
<td>Trust-Region Method</td>
</tr>
<tr>
<td>NEWRAP</td>
<td>Newton-Raphson Method with Line Search</td>
</tr>
<tr>
<td>NRRIDG</td>
<td>Newton-Raphson Method with Ridging</td>
</tr>
<tr>
<td>QUANEW</td>
<td>Quasi-Newton Methods (DBFGS, DDFP, BFGS, DFP)</td>
</tr>
<tr>
<td>DBLDOG</td>
<td>Double-Dogleg Method (DBFGS, DDFP)</td>
</tr>
<tr>
<td>CONGRA</td>
<td>Conjugate Gradient Methods (PB, FR, PR, CD)</td>
</tr>
</tbody>
</table>

Table 17.14 shows, for each optimization technique, which derivatives are needed (first-order or second-order) and what kind of constraints (boundary, linear, or nonlinear) can be imposed on the parameters.

\textbf{Table 17.14.} Derivatives Needed and Constraints Allowed

<table>
<thead>
<tr>
<th>TECH=</th>
<th>Derivatives</th>
<th>Constraints</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>First Order</td>
<td>Second Order</td>
</tr>
<tr>
<td>LEVMAR</td>
<td>x</td>
<td>x</td>
</tr>
<tr>
<td>TRUREG</td>
<td>x</td>
<td>x</td>
</tr>
<tr>
<td>NEWRAP</td>
<td>x</td>
<td>x</td>
</tr>
<tr>
<td>NRRIDG</td>
<td>x</td>
<td>x</td>
</tr>
<tr>
<td>QUANEW</td>
<td>x</td>
<td>-</td>
</tr>
<tr>
<td>DBLDOG</td>
<td>x</td>
<td>-</td>
</tr>
<tr>
<td>CONGRA</td>
<td>x</td>
<td>-</td>
</tr>
</tbody>
</table>

The Levenberg-Marquardt, trust-region, and Newton-Raphson techniques are usually the most reliable, work well with boundary and general linear constraints, and generally converge after a few iterations to a precise solution. However, these techniques need to compute a Hessian matrix in each iteration. For HESSALG=1, this means that you need about \(4(n(n+1)/2)t\) bytes of work memory \((n = \) the number of manifest variables, \(t = \) the number of parameters to estimate) to store the Jacobian and its cross product. With HESSALG=2 or HESSALG=3, you do not need this work memory, but the use of a utility file increases execution time. Computing the approximate Hessian in each iteration can be very time- and memory-consuming, especially for large problems (more than 60 or 100 parameters, depending on the computer used). For large problems, a quasi-Newton technique, especially with the BFGS update, can be far more efficient.

For a poor choice of initial values, the Levenberg-Marquardt method seems to be more reliable.

If memory problems occur, you can use one of the conjugate gradient techniques, but they are generally slower and less reliable than the methods that use second-order information.
There are several options to control the optimization process. First of all, you can specify various termination criteria. You can specify the GCONV= option to specify a relative gradient termination criterion. If there are active boundary constraints, only those gradient components that correspond to inactive constraints contribute to the criterion. When you want very precise parameter estimates, the GCONV= option is useful. Other criteria that use relative changes in function values or parameter estimates in consecutive iterations can lead to early termination when active constraints cause small steps to occur. The small default value for the FCONV= option helps prevent early termination. Using the MAXITER= and MAXFUNC= options enables you to specify the maximum number of iterations and function calls in the optimization process. These limits are especially useful in combination with the INRAM= and OUTRAM= options; you can run a few iterations at a time, inspect the results, and decide whether to continue iterating.

**Nonlinearly Constrained QN Optimization**

The algorithm used for nonlinearly constrained quasi-Newton optimization is an efficient modification of Powell’s (1978a, 1978b, 1982a, 1982b) Variable Metric Constrained WatchDog (VMCWD) algorithm. A similar but older algorithm (VF02AD) is part of the Harwell library. Both VMCWD and VF02AD use Fletcher’s VE02AD algorithm (also part of the Harwell library) for positive definite quadratic programming. The PROC CALIS QUANEW implementation uses a quadratic programming subroutine that updates and downdates the approximation of the Cholesky factor when the active set changes. The nonlinear QUANEW algorithm is not a feasible point algorithm, and the value of the objective function need not decrease (minimization) or increase (maximization) monotonically. Instead, the algorithm tries to reduce a linear combination of the objective function and constraint violations, called the **merit function**.

The following are similarities and differences between this algorithm and VMCWD:

- A modification of this algorithm can be performed by specifying VERSION=1, which replaces the update of the Lagrange vector \( \mu \) with the original update of Powell (1978a, 1978b), which is used in VF02AD. This can be helpful for some applications with linearly dependent active constraints.
- If the VERSION= option is not specified or VERSION=2 is specified, the evaluation of the Lagrange vector \( \mu \) is performed in the same way as Powell (1982a, 1982b) describes.
- Instead of updating an approximate Hessian matrix, this algorithm uses the dual BFGS (or DFP) update that updates the Cholesky factor of an approximate Hessian. If the condition of the updated matrix gets too bad, a restart is done with a positive diagonal matrix. At the end of the first iteration after each restart, the Cholesky factor is scaled.
- The Cholesky factor is loaded into the quadratic programming subroutine, automatically ensuring positive definiteness of the problem. During the quadratic programming step, the Cholesky factor of the projected Hessian matrix \( Z_k^T G Z_k \) and the \( QT \) decomposition are updated simultaneously when the active set changes. Refer to Gill et al. (1984) for more information.
Use of Optimization Techniques

- The line-search strategy is very similar to that of Powell (1982a, 1982b). However, this algorithm does not call for derivatives during the line search; hence, it generally needs fewer derivative calls than function calls. The VMCWD algorithm always requires the same number of derivative and function calls. It was also found in several applications of VMCWD that Powell’s line-search method sometimes uses steps that are too long during the first iterations. In those cases, you can use the INSTEP= option specification to restrict the step length \( \alpha \) of the first iterations.

- Also the watchdog strategy is similar to that of Powell (1982a, 1982b). However, this algorithm doesn’t return automatically after a fixed number of iterations to a former better point. A return here is further delayed if the observed function reduction is close to the expected function reduction of the quadratic model.

- Although Powell’s termination criterion still is used (as FCONV2), the QUANEW implementation uses two additional termination criteria (GCONV and ABSGCONV).

This algorithm is automatically invoked when you specify the NLINCON statement. The nonlinear QUANEW algorithm needs the Jacobian matrix of the first-order derivatives (constraints normals) of the constraints

\[
(\nabla c_i) = \left( \frac{\partial c_i}{\partial x_j} \right), \quad i = 1, \ldots, nc, j = 1, \ldots, n
\]

where \( nc \) is the number of nonlinear constraints for a given point \( x \).

You can specify two update formulas with the UPDATE= option:

- UPDATE=DBFGS performs the dual BFGS update of the Cholesky factor of the Hessian matrix. This is the default.
- UPDATE=DDFP performs the dual DFP update of the Cholesky factor of the Hessian matrix.

This algorithm uses its own line-search technique. All options and parameters (except the INSTEP= option) controlling the line search in the other algorithms do not apply here. In several applications, large steps in the first iterations are troublesome. You can specify the INSTEP= option to impose an upper bound for the step size \( \alpha \) during the first five iterations. The values of the LCSINGULAR=, LCEPSILON=, and LCDEACT= options, which control the processing of linear and boundary constraints, are valid only for the quadratic programming subroutine used in each iteration of the nonlinear constraints QUANEW algorithm.

**Optimization and Iteration History**

The optimization and iteration histories are displayed by default because it is important to check for possible convergence problems.

The optimization history includes the following summary of information about the initial state of the optimization:
• the number of constraints that are active at the starting point, or more precisely, the number of constraints that are currently members of the working set. If this number is followed by a plus sign, there are more active constraints, of which at least one is temporarily released from the working set due to negative Lagrange multipliers.

• the value of the objective function at the starting point

• if the (projected) gradient is available, the value of the largest absolute (projected) gradient element

• for the TRUREG and LEVMAR subroutines, the initial radius of the trust region around the starting point

The optimization history ends with some information concerning the optimization result:

• the number of constraints that are active at the final point, or more precisely, the number of constraints that are currently members of the working set. If this number is followed by a plus sign, there are more active constraints, of which at least one is temporarily released from the working set due to negative Lagrange multipliers.

• the value of the objective function at the final point

• if the (projected) gradient is available, the value of the largest absolute (projected) gradient element

• other information specific to the optimization technique

The iteration history generally consists of one line of displayed output containing the most important information for each iteration. The _LIST_ variable (see the “SAS Program Statements” section on page 474) also enables you to display the parameter estimates and the gradient in some or all iterations.

The iteration history always includes the following (the words in parentheses are the column header output):

• the iteration number (Iter)
• the number of iteration restarts (rest)
• the number of function calls (nfun)
• the number of active constraints (act)
• the value of the optimization criterion (optcrit)
• the difference between adjacent function values (difcrit)
• the maximum of the absolute gradient components corresponding to inactive boundary constraints (maxgrad)

An apostrophe trailing the number of active constraints indicates that at least one of the active constraints is released from the active set due to a significant Lagrange multiplier.
For the Levenberg-Marquardt technique (LEVMAR), the iteration history also includes the following information:

- An asterisk trailing the iteration number means that the computed Hessian approximation is singular and consequently ridged with a positive lambda value. If all or the last several iterations show a singular Hessian approximation, the problem is not sufficiently identified. Thus, there are other locally optimal solutions that lead to the same optimum function value for different parameter values. This implies that standard errors for the parameter estimates are not computable without the addition of further constraints.

- The value of the Lagrange multiplier (lambda); this is 0 if the optimum of the quadratic function approximation is inside the trust region (a trust-region-scaled Newton step can be performed) and is greater than 0 when the optimum of the quadratic function approximation is located at the boundary of the trust region (the scaled Newton step is too long to fit in the trust region and a quadratic constraint optimization is performed). Large values indicate optimization difficulties. For a nonsingular Hessian matrix, the value of lambda should go to 0 during the last iterations, indicating that the objective function can be well approximated by a quadratic function in a small neighborhood of the optimum point. An increasing lambda value often indicates problems in the optimization process.

- The value of the ratio \(\rho\) between the actually achieved difference in function values and the predicted difference in the function values on the basis of the quadratic function approximation. Values much less than 1 indicate optimization difficulties. The value of the ratio \(\rho\) indicates the goodness of the quadratic function approximation; in other words, \(\rho << 1\) means that the radius of the trust region has to be reduced. A fairly large value of \(\rho\) means that the radius of the trust region need not be changed. And a value close to or larger than 1 means that the radius can be increased, indicating a good quadratic function approximation.

For the Newton-Raphson technique (NRRIDG), the iteration history also includes the following information:

- The value of the ridge parameter. This is 0 when a Newton step can be performed, and it is greater than 0 when either the Hessian approximation is singular or a Newton step fails to reduce the optimization criterion. Large values indicate optimization difficulties.

- The value of the ratio \(\rho\) (rho) between the actually achieved difference in function values and the predicted difference in the function values on the basis of the quadratic function approximation. Values much less than 1.0 indicate optimization difficulties.

For the Newton-Raphson with line-search technique (NEWRAP), the iteration history also includes:

- The step size \(\alpha\) (alpha) computed with one of the line-search algorithms
the slope of the search direction at the current parameter iterate. For minimization, this value should be significantly negative. Otherwise, the line-search algorithm has difficulty reducing the function value sufficiently.

For the Trust-Region technique (TRUREG), the iteration history also includes the following information:

- An asterisk after the iteration number means that the computed Hessian approximation is singular and consequently ridged with a positive lambda value.
- the value of the Lagrange multiplier (lambda). This value is zero when the optimum of the quadratic function approximation is inside the trust region (a trust-region-scaled Newton step can be performed) and is greater than zero when the optimum of the quadratic function approximation is located at the boundary of the trust region (the scaled Newton step is too long to fit in the trust region and a quadratically constrained optimization is performed). Large values indicate optimization difficulties. As in Gay (1983), a negative lambda value indicates the special case of an indefinite Hessian matrix (the smallest eigenvalue is negative in minimization).
- the value of the radius $\Delta$ of the trust region. Small trust region radius values combined with large lambda values in subsequent iterations indicate optimization problems.

For the quasi-Newton (QUANEW) and conjugate gradient (CONGRA) techniques, the iteration history also includes the following information:

- the step size (alpha) computed with one of the line-search algorithms
- the descent of the search direction at the current parameter iterate. This value should be significantly smaller than 0. Otherwise, the line-search algorithm has difficulty reducing the function value sufficiently.

Frequent update restarts (rest) of a quasi-Newton algorithm often indicate numerical problems related to required properties of the approximate Hessian update, and they decrease the speed of convergence. This can happen particularly if the ABSGCONV= termination criterion is too small, that is, when the requested precision cannot be obtained by quasi-Newton optimization. Generally, the number of automatic restarts used by conjugate gradient methods are much higher.

For the nonlinearly constrained quasi-Newton technique, the iteration history also includes the following information:

- the maximum value of all constraint violations,
  \[
  \text{conmax} = \max(|c_i(x)| : c_i(x) < 0)
  \]
- the value of the predicted function reduction used with the GCONV and FCONV2 termination criteria,
  \[
  \text{pred} = |g(x^{(k)})s(x^{(k)})| + \sum_{i=1}^{m} |\lambda_i c_i(x^{(k)})|
  \]
• the step size $\alpha$ of the quasi-Newton step. Note that this algorithm works with a special line-search algorithm.

• the maximum element of the gradient of the Lagrange function,

$$\text{Ifgmax} = \nabla_x L(x^{(k)}, \lambda^{(k)}) = \nabla_x f(x^{(k)}) - \sum_{i=1}^{m} \lambda_i^{(k)} \nabla_x c_i(x^{(k)})$$

For the double dogleg technique, the iteration history also includes the following information:

• the parameter $\lambda$ of the double-dogleg step. A value $\lambda = 0$ corresponds to the full (quasi) Newton step.

• the slope of the search direction at the current parameter iterate. For minimization, this value should be significantly negative.

### Line-Search Methods

In each iteration $k$, the (dual) quasi-Newton, hybrid quasi-Newton, conjugate gradient, and Newton-Raphson minimization techniques use iterative line-search algorithms that try to optimize a linear, quadratic, or cubic approximation of the nonlinear objective function $f$ of $n$ parameters $x$ along a feasible descent search direction $s^{(k)}$

$$f(x^{(k+1)}) = f(x^{(k)} + \alpha^{(k)} s^{(k)})$$

by computing an approximately optimal scalar $\alpha^{(k)} > 0$. Since the outside iteration process is based only on the approximation of the objective function, the inside iteration of the line-search algorithm does not have to be perfect. Usually, it is satisfactory that the choice of $\alpha$ significantly reduces (in a minimization) the objective function. Criteria often used for termination of line-search algorithms are the Goldstein conditions (Fletcher 1987).

Various line-search algorithms can be selected by using the LIS= option (page 427). The line-search methods LIS=1, LIS=2, and LIS=3 satisfy the left-hand-side and right-hand-side Goldstein conditions (refer to Fletcher 1987). When derivatives are available, the line-search methods LIS=6, LIS=7, and LIS=8 try to satisfy the right-hand-side Goldstein condition; if derivatives are not available, these line-search algorithms use only function calls.

The line-search method LIS=2 seems to be superior when function evaluation consumes significantly less computation time than gradient evaluation. Therefore, LIS=2 is the default value for Newton-Raphson, (dual) quasi-Newton, and conjugate gradient optimizations.

### Restricting the Step Length

Almost all line-search algorithms use iterative extrapolation techniques that can easily lead to feasible points where the objective function $f$ is no longer defined (resulting in indefinite matrices for ML estimation) or is difficult to compute (resulting in
floating point overflows). Therefore, PROC CALIS provides options that restrict the step length or trust region radius, especially during the first main iterations.

The inner product $g^T s$ of the gradient $g$ and the search direction $s$ is the slope of $f(\alpha) = f(x + \alpha s)$ along the search direction $s$ with step length $\alpha$. The default starting value $\alpha^{(0)} = \alpha^{(k,0)}$ in each line-search algorithm ($\min_{\alpha > 0} f(x + \alpha s)$) during the main iteration $k$ is computed in three steps.

1. Use either the difference $df = |f^{(k)} - f^{(k-1)}|$ of the function values during the last two consecutive iterations or the final stepsize value $\alpha^-$ of the previous iteration $k - 1$ to compute a first value $\alpha^{(0)}_1$.
   - Using the DAMPSTEP<=r> option:
     \[ \alpha^{(0)}_1 = \min(1, r \alpha^-) \]
     The initial value for the new step length can be no larger than $r$ times the final step length $\alpha^-$ of the previous iteration. The default is $r = 2$.
   - Not using the DAMPSTEP option:
     \[ \alpha^{(0)}_1 = \begin{cases} 
     \text{step} & \text{if } 0.1 \leq \text{step} \leq 10 \\
     10 & \text{if } \text{step} > 10 \\
     0.1 & \text{if } \text{step} < 0.1 
     \end{cases} \]
     with
     \[ \text{step} = \begin{cases} 
     \frac{df}{|g^T s|} & \text{if } |g^T s| \geq \epsilon \max(100df, 1) \\
     1 & \text{otherwise} 
     \end{cases} \]
     This value of $\alpha^{(0)}_1$ can be too large and can lead to a difficult or impossible function evaluation, especially for highly nonlinear functions such as the EXP function.

2. During the first five iterations, the second step enables you to reduce $\alpha^{(0)}_1$ to a smaller starting value $\alpha^{(0)}_2$ using the INSTEP=r option:
   \[ \alpha^{(0)}_2 = \min(\alpha^{(0)}_1, r) \]
   After more than five iterations, $\alpha^{(0)}_2$ is set to $\alpha^{(0)}_1$.

3. The third step can further reduce the step length by
   \[ \alpha^{(0)}_3 = \min(\alpha^{(0)}_2, \min(10, u)) \]
   where $u$ is the maximum length of a step inside the feasible region.

The INSTEP=r option lets you specify a smaller or larger radius of the trust region used in the first iteration by the trust-region, double-dogleg, and Levenberg-Marquardt algorithm. The default initial trust region radius is the length of the scaled gradient (Moré 1978). This step corresponds to the default radius factor of $r = 1$. This choice is successful in most practical applications of the TRUREG, DBLDOG, and LEVMAR algorithms. However, for bad initial values used in the analysis of a
covariance matrix with high variances, or for highly nonlinear constraints (such as using the EXP function) in your programming code, the default start radius can result in arithmetic overflows. If this happens, you can try decreasing values of INSTEP=r, 0 < r < 1, until the iteration starts successfully. A small factor r also affects the trust region radius of the next steps because the radius is changed in each iteration by a factor 0 < c ≤ 4 depending on the ρ ratio. Reducing the radius corresponds to increasing the ridge parameter λ that produces smaller steps directed closer toward the gradient direction.

Modification Indices

While fitting structural models, you may want to modify the specified model in order to

• reduce the χ² value significantly
• reduce the number of parameters to estimate without increasing the χ² value too much

If you specify the MODIFICATION or MOD option, PROC CALIS computes and displays a default set of modification indices:

• **Univariate Lagrange multiplier test indices** for most elements in the model matrices that are constrained to equal constants. These are second-order approximations of the decrease in the χ² value that would result from allowing the constant matrix element to vary. Besides the value of the Lagrange multiplier, the corresponding probability (df = 1) and the approximate change of the parameter value (should the constant be changed to a parameter) are displayed. If allowing the constant to be a free estimated parameter would result in a singular information matrix, the string ‘sing’ is displayed instead of the Lagrange multiplier index. Not all elements in the model matrices should be allowed to vary; the diagonal elements of the inverse matrices in the RAM or LINEQS model must be constant ones. The univariate Lagrange multipliers are displayed at the constant locations of the model matrices.

• **Univariate Wald test indices** for those matrix elements that correspond to parameter estimates in the model. These are second-order approximations of the increase in the χ² value that would result from constraining the parameter to a 0 constant. The univariate Wald test indices are the same as the t values that are displayed together with the parameter estimates and standard errors. The univariate Wald test indices are displayed at the parameter locations of the model matrices.

• **Univariate Lagrange multiplier test indices** that are second-order approximations of the decrease in the χ² value that would result from the release of equality constraints. Multiple equality constraints containing n > 2 parameters are tested successively in n steps, each assuming the release of one of the equality-constrained parameters. The expected change of the parameter values of the separated parameter and the remaining parameter cluster are displayed, too.
Chapter 17. The CALIS Procedure

- **Univariate Lagrange multiplier test indices** for releasing *active boundary constraints* specified by the BOUNDS statement.

- **Stepwise multivariate Wald test indices** for constraining estimated parameters to 0 are computed and displayed. In each step, the parameter that would lead to the smallest increase in the multivariate $\chi^2$ value is set to 0. Besides the multivariate $\chi^2$ value and its probability, the univariate increments are also displayed. The process stops when the univariate probability is smaller than the specified value in the SLMW= option.

All of the preceding tests are approximations. You can often get more accurate tests by actually fitting different models and computing likelihood ratio tests. For more details about the Wald and the Lagrange multiplier test, refer to MacCallum (1986), Buse (1982), Bentler (1986), or Lee (1985).

Note that, for large model matrices, the computation time for the default modification indices can considerably exceed the time needed for the minimization process.

The modification indices are not computed for unweighted least-squares or diagonally weighted least-squares estimation.

**CAUTION:** Modification indices are not computed if the model matrix is an identity matrix (IDE or ZID), a selection matrix (PER), or the first matrix $J$ in the LINEQS model. If you want to display the modification indices for such a matrix, you should specify the matrix as another type; for example, specify an identity matrix used in the COSAN statement as a diagonal matrix with constant diagonal elements of 1.

### Constrained Estimation Using Program Code

The CALIS procedure offers a very flexible way to constrain parameter estimates. You can use your own programming statements to express special properties of the parameter estimates. This tool is also present in McDonald’s COSAN implementation but is considerably easier to use in the CALIS procedure. PROC CALIS is able to compute analytic first- and second-order derivatives that you would have to specify using the COSAN program. There are also three PROC CALIS statements you can use:

- the BOUNDS statement, to specify simple bounds on the parameters used in the optimization process
- the LINCON statement, to specify general linear equality and inequality constraints on the parameters used in the optimization process
- the NLINCON statement, to specify general nonlinear equality and inequality constraints on the parameters used in the optimization process. The variables listed in the NLINCON statement must be specified in the program code.

There are some traditional ways to enforce parameter constraints by using parameter transformations (McDonald 1980):

- **One-sided boundary constraints:** For example, the parameter $q_k$ should be at
least as large (or at most as small) as a given constant value \( a \) (or \( b \)),

\[
q_k \geq a \quad \text{or} \quad q_k \leq b
\]

This inequality constraint can be expressed as an equality constraint

\[
q_k = a + x_j^2 \quad \text{or} \quad q_k = b - x_j^2
\]

in which the fundamental parameter \( x_j \) is unconstrained.

- **Two-sided boundary constraints:** For example, the parameter \( q_k \) should be located between two given constant values \( a \) and \( b \), \( a < b \),

\[
a \leq q_k \leq b
\]

This inequality constraint can be expressed as an equality constraint

\[
q_k = a + b \cdot \frac{\exp(x_j)}{1 + \exp(x_j)}
\]

in which the fundamental parameter \( x_j \) is unconstrained.

- **One-sided order constraints:** For example, the parameters \( q_1, \ldots, q_k \) should be ordered in the form

\[
q_1 \leq q_2, \quad q_1 \leq q_3, \quad \ldots, \quad q_1 \leq q_k
\]

These inequality constraints can be expressed as a set of equality constraints

\[
q_1 = x_1, \quad q_2 = x_1 + x_2^2, \quad \ldots, \quad q_k = x_1 + x_k^2
\]

in which the fundamental parameters \( x_1, \ldots, x_k \) are unconstrained.

- **Two-sided order constraints:** For example, the parameters \( q_1, \ldots, q_k \) should be ordered in the form

\[
q_1 \leq q_2 \leq q_3 \leq \ldots \leq q_k
\]

These inequality constraints can be expressed as a set of equality constraints

\[
q_1 = x_1, \quad q_2 = q_1+x_2^2, \quad \ldots, \quad q_k = q_{k-1} + x_k^2
\]

in which the fundamental parameters \( x_1, \ldots, x_k \) are unconstrained.

- **Linear equation constraints:** For example, the parameters \( q_1, q_2, q_3 \) should be linearly constrained in the form

\[
q_1 + q_2 + q_3 = a
\]

which can be expressed in the form of three explicit equations

\[
q_1 = x_1, \quad q_2 = x_2, \quad q_3 = a - x_1 - x_2
\]

in which the fundamental parameters \( x_1 \) and \( x_2 \) are unconstrained.
Refer to McDonald (1980) and Browne (1982) for further notes on reparameterizing techniques. If the optimization problem is not too large to apply the Levenberg-Marquardt or Newton-Raphson algorithm, boundary constraints should be requested by the BOUNDS statement rather than by reparameterizing code. If the problem is so large that you must use a quasi-Newton or conjugate gradient algorithm, reparameterizing techniques may be more efficient than the BOUNDS statement.

### Counting the Degrees of Freedom

In a regression problem, the number of degrees of freedom for the error estimate is the number of observations in the data set minus the number of parameters. The NOBS=, DFR= (RDF=), and DFE= (EDF=) options refer to degrees of freedom in this sense. However, these values are not related to the degrees of freedom of a test statistic used in a covariance or correlation structure analysis. The NOBS=, DFR=, and DFE= options should be used in PROC CALIS to specify only the effective number of observations in the input DATA= data set.

In general, the number of degrees of freedom in a covariance or correlation structure analysis is defined as the difference between the number of nonredundant values $q$ in the observed $n \times n$ correlation or covariance matrix $S$ and the number $t$ of free parameters $X$ used in the fit of the specified model, $df = q - t$. Both values, $q$ and $t$, are counted differently in different situations by PROC CALIS.

The number of nonredundant values $q$ is generally equal to the number of lower triangular elements in the $n \times n$ moment matrix $S$ including all diagonal elements, minus a constant $c$ dependent upon special circumstances,

$$ q = n(n + 1)/2 - c $$

The number $c$ is evaluated by adding the following quantities:

- If you specify a linear structural equation model containing exogenous manifest variables by using the RAM or LINEQS statement, PROC CALIS adds to $c$ the number of variances and covariances among these manifest exogenous variables, which are automatically set in the corresponding locations of the central model matrices (see the section “Exogenous Manifest Variables” on page 508).
- If you specify the DFREDUCE=i option, PROC CALIS adds the specified number $i$ to $c$. The number $i$ can be a negative integer.
- If you specify the NODIAG option to exclude the fit of the diagonal elements of the data matrix $S$, PROC CALIS adds the number $n$ of diagonal elements to $c$.
- If all the following conditions hold, then PROC CALIS adds to $c$ the number of the diagonal locations:
  - NODIAG and DFREDUCE= options are not specified.
  - A correlation structure is being fitted.
  - The predicted correlation matrix contains constants on the diagonal.
In some complicated models, especially those using programming statements, PROC CALIS may not be able to detect all the constant predicted values. In such cases, you must specify the DFREDUCE= option to get the correct degrees of freedom.

The number \( t \) is the number of different parameter names used in constructing the model if you do not use programming statements to impose constraints on the parameters. Using programming statements in general introduces two kinds of parameters:

- independent parameters, which are used only at the right-hand side of the expressions
- dependent parameters, which are used at least once at the left-hand side of the expressions

The independent parameters belong to the parameters involved in the estimation process, whereas the dependent parameters are fully defined by the programming statements and can be computed from the independent parameters. In this case, the number \( t \) is the number of different parameter names used in the model specification, but not used in the programming statements, plus the number of independent parameters. The independent parameters and their initial values can be defined in a model specification statement or in a PARMS statement.

---

**Computational Problems**

**First Iteration Overflows**

Analyzing a covariance matrix including high variances in the diagonal and using bad initial estimates for the parameters can easily lead to arithmetic overflows in the first iterations of the minimization algorithm. The line-search algorithms that work with cubic extrapolation are especially sensitive to arithmetic overflows. If this occurs with quasi-Newton or conjugate gradient minimization, you can specify the INSTEP= option to reduce the length of the first step. If an arithmetic overflow occurs in the first iteration of the Levenberg-Marquardt algorithm, you can specify the INSTEP= option to reduce the trust region radius of the first iteration. You also can change the minimization technique or the line-search method. If none of these help, you should consider

- scaling the covariance matrix
- providing better initial values
- changing the model

**No Convergence of Minimization Process**

If convergence does not occur during the minimization process, perform the following tasks:

- If there are negative variance estimates in the diagonal locations of the central model matrices, you can
  
  - specify the BOUNDS statement to obtain nonnegative variance estimates
specify the HEYWOOD option, if the FACTOR model statement is specified.

- Change the estimation method to obtain a better set of initial estimates. For example, if you use METHOD=ML, you can
  - change to METHOD=LSML
  - run some iterations with METHOD=DWLS or METHOD=GLS, write the results in an OUTRAM= data set, and use the results as initial values specified by an INRAM= data set in a second run with METHOD=ML

- Change the optimization technique. For example, if you use the default TECH=LEVMAR, you can
  - change to TECH=QUANEW or to TECH=NEWRAP
  - run some iterations with TECH=CONGRA, write the results in an OUTRAM= data set, and use the results as initial values specified by an INRAM= data set in a second run with a different TECH=

- Change or modify the update technique or the line-search algorithm, or both, when using TECH=QUANEW or TECH=CONGRA. For example, if you use the default update formula and the default line-search algorithm, you can
  - change the update formula with the UPDATE= option
  - change the line-search algorithm with the LIS= option
  - specify a more precise line search with the LSPRECISION= option, if you use LIS=2 or LIS=3

- You can allow more iterations and function calls by using the MAXIT= and MAXFU= options.

- Change the initial values. For many categories of model specifications done by the LINEQS, RAM, or FACTOR model, PROC CALIS computes an appropriate set of initial values automatically. However, for some of the model specifications (for example, structural equations with latent variables on the left-hand side and manifest variables on the right-hand side), PROC CALIS can generate very obscure initial values. In these cases, you have to set the initial values yourself.
  - Increase the initial values of the parameters located at the diagonal of central matrices
    * manually, by setting the values in the model specification
    * automatically, by using the DEMPHAS= option
  - Use a slightly different, but more stable, model to obtain preliminary estimates.
  - Use additional information to specify initial values, for example, by using other SAS software like the FACTOR, REG, SYSLIN, and MODEL (SYSNLIN) procedures for the modified, unrestricted model case.

- Change the optimization technique. For example, if you use the default TECH=LEVMAR, you can
Compute the Problems

- change to TECH=QUANEW or to TECH=NEWRAP
- run some iterations with TECH=CONGRA, write the results in an OUT- RAM= data set, and use the results as initial values specified by an IN- RAM= data set in a second run with a different TECH= technique

- Change or modify the update technique or the line-search algorithm, or both, when using TECH=QUANEW or TECH=CONGRA. For example, if you use the default update formula and the default line-search algorithm, you can
  - change the update formula with the UPDATE= option
  - change the line-search algorithm with the LIS= option
  - specify a more precise line search with the LSPRECISION= option, if you use LIS=2 or LIS=3

- Temporarily change the estimation method to obtain a better set of initial estimates. For example, if you use METHOD=ML, you can
  - change to METHOD=LSML
  - run some iterations with METHOD=DWLS or GLS, write the results in an OUTRAM= data set, and use the results as initial values specified by an INRAM= data set in a second run with METHOD=ML

- You can allow more iterations and function calls by using the MAXIT= and MAXFU= options.

Unidentified Model

The parameter vector $\mathbf{x}$ in the covariance structure model

$$
\mathbf{C} = \mathbf{C}(\mathbf{x})
$$

is said to be identified in a parameter space $G$, if

$$
\mathbf{C}(\mathbf{x}) = \mathbf{C}(\tilde{\mathbf{x}}), \quad \tilde{\mathbf{x}} \in G
$$

implies $\mathbf{x} = \tilde{\mathbf{x}}$. The parameter estimates that result from an unidentified model can be very far from the parameter estimates of a very similar but identified model. They are usually machine dependent. Don’t use parameter estimates of an unidentified model as initial values for another run of PROC CALIS.

Singular Predicted Model Matrix

You can easily specify models with singular predicted model matrices, for example, by fixing diagonal elements of central matrices to 0. In such cases, you cannot compute maximum likelihood estimates (the ML function value $F$ is not defined). Since singular predicted model matrices can also occur temporarily in the minimization process, PROC CALIS tries in such cases to change the parameter estimates so that the predicted model matrix becomes positive definite. In such cases, the following message is displayed:

**NOTE: Parameter set changed.**
This process does not always work well, especially if there are fixed instead of variable diagonal elements in the central model matrices. A famous example where you cannot compute ML estimates is a component analysis with fewer components than given manifest variables. See the section “FACTOR Model Statement” on page 453 for more details. If you continue to get a singular predicted model matrix after changing initial values and optimization techniques, then your model is perhaps specified so that ML estimates cannot be computed.

**Saving Computing Time**

For large models, the most computing time is needed to compute the modification indices. If you don’t really need the Lagrange multipliers or multiple Wald test indices (the univariate Wald test indices are the same as the t values), using the NOMOD option can save a considerable amount of computing time.

**Central Matrices with Negative Eigenvalues**

A covariance matrix cannot have negative eigenvalues, since a negative eigenvalue means that some linear combination of the variables has negative variance. PROC CALIS displays a warning if a central model matrix has negative eigenvalues but does not actually compute the eigenvalues. Sometimes this warning can be triggered by 0 or very small positive eigenvalues that appear negative because of numerical error. If you want to be sure that the central model matrix you are fitting can be considered to be a variance-covariance matrix, you can use the SAS/IML command VAL=EIGVAL(U) to compute the vector VAL of eigenvalues of matrix U.

**Negative $R^2$ Values**

The estimated squared multiple correlations $R^2$ of the endogenous variables are computed using the estimated error variances

\[
R^2_i = 1 - \frac{\text{var}(\zeta_i)}{\text{var}(\eta_i)}
\]

If the model is a poor fit, it is possible that $\text{var}(\zeta_i) > \text{var}(\eta_i)$, which results in $R^2_i < 0$.

**Displayed Output**

The output displayed by PROC CALIS depends on the statement used to specify the model. Since an analysis requested by the LINEQS or RAM statement implies the analysis of a structural equation model, more statistics can be computed and displayed than for a covariance structure analysis following the generalized COSAN model requested by the COSAN statement. The displayed output resulting from use of the FACTOR statement includes all the COSAN displayed output as well as more statistics displayed only when you specify the FACTOR statement. Since the displayed output using the RAM statement differs only in its form from that generated by the LINEQS statement, in this section distinctions are made between COSAN and LINEQS output only.

The unweighted least-squares and diagonally weighted least-squares estimation methods do not provide a sufficient statistical basis to provide the following output.
(neither displayed nor written to an OUTEST= data set):

- most of the fit indices
- approximate standard errors
- normalized or asymptotically standardized residuals
- modification indices
- information matrix
- covariance matrix of parameter estimates

The notation $S = (s_{ij})$ is used for the analyzed covariance or correlation matrix, $C = (c_{ij})$ for the predicted model matrix, $W$ for the weight matrix (for example, $W = I$ for ULS, $W = S$ for GLS, $W = C$ for ML estimates), $X$ for the vector of optimal parameter estimates, $n$ for the number of manifest variables, $t$ for the number of parameter estimates, and $N$ for the sample size.

The output of PROC CALIS includes

- COSAN and LINEQS: List of the matrices and their properties specified by the generalized COSAN model if you specify at least the PSHORT option.
- LINEQS: List of manifest variables that are not used in the specified model and that are automatically omitted from the analysis. Note that there is no automatic variable reduction with the COSAN or FACTOR statement. If necessary, you should use the VAR statement in these cases.
- LINEQS: List of the endogenous and exogenous variables specified by the LINEQS, STD, and COV statements if you specify at least the PSHORT option.
- COSAN: Initial values of the parameter matrices indicating positions of constants and parameters. The output, or at least the default output, is displayed if you specify the PINITIAL option.
- LINEQS: The set of structural equations containing the initial values and indicating constants and parameters, and output of the initial error variances and covariances. The output, or at least the default output, is displayed if you specify the PINITIAL option.
- COSAN and LINEQS: The weight matrix $W$ is displayed if GLS, WLS, or DWLS estimation is used and you specify the PWEIGHT or PALL option.
- COSAN and LINEQS: General information about the estimation problem: number of observations ($N$), number of manifest variables ($n$), amount of independent information in the data matrix (information, $n(n + 1)/2$), number of terms and matrices in the specified generalized COSAN model, and number of parameters to be estimated (parameters, $t$). If there are no exogenous manifest variables, the difference between the amount of independent information ($n(n + 1)/2$) and the number of requested estimates ($t$) is equal to the degrees of freedom ($df$). A necessary condition for a model to be identified is that the
degrees of freedom are nonnegative. The output, or at least the default output, is displayed if you specify the SIMPLE option.

- **COSAN and LINEQS**: Mean and Std Dev (standard deviation) of each variable if you specify the SIMPLE option, as well as skewness and kurtosis if the DATA= data set is a raw data set and you specify the KURTOSIS option.

- **COSAN and LINEQS**: Various coefficients of multivariate kurtosis and the numbers of observations that contribute most to the normalized multivariate kurtosis if the DATA= data set is a raw data set and the KURTOSIS option, or you specify at least the PRINT option. See the section “Measures of Multivariate Kurtosis” on page 503 for more information.

- **COSAN and LINEQS**: Covariance or correlation matrix to be analyzed and the value of its determinant if you specify the output option PCORR or PALL. A 0 determinant indicates a singular data matrix. In this case, the generalized least-squares estimates with default weight matrix $S$ and maximum likelihood estimates cannot be computed.

- **LINEQS**: If exogenous manifest variables in the linear structural equation model are specified, then there is a one-to-one relationship between the given covariances and corresponding estimates in the central model matrix $\Phi$ or $P$. The output indicates which manifest variables are recognized as exogenous, that is, for which variables the entries in the central model matrix are set. The output, or at least the default output, is displayed if you specify the PINITIAL option.

- **COSAN and LINEQS**: Vector of parameter names, initial values, and corresponding matrix locations, also indicating dependent variable names used in your program statements that are not allocated to matrix locations and have no influence on the fit function. The output, or at least the default output, is displayed if you specify the PINITIAL option.

- **COSAN and LINEQS**: The pattern of variable and constant elements of the predicted moment matrix that is predetermined by the analysis model is displayed if there are significant differences between constant elements in the predicted model matrix and the data matrix and you specify at least the PSHORT option. It is also displayed if you specify the PREDET option. The output indicates the differences between constant values in the predicted model matrix and the data matrix that is analyzed.

- **COSAN and LINEQS**: Special features of the optimization technique chosen if you specify at least the PSHORT option.

- **COSAN and LINEQS**: Optimization history if at least the PSHORT option is specified. For more details, see the section “Use of Optimization Techniques” on page 510.

- **COSAN and LINEQS**: Specific output requested by options in the NLOPTIONS statement; for example, parameter estimates, gradient, gradient of Lagrange function, constraints, Lagrange multipliers, projected gradient, Hessian, projected Hessian, Hessian of Lagrange function, Jacobian of nonlinear constraints.
• COSAN and LINEQS: The predicted model matrix and its determinant, if you specify the output option PCORR or PALL.

• COSAN and LINEQS: Residual and normalized residual matrix if you specify the RESIDUAL, or at least the PRINT option. The variance standardized or asymptotically standardized residual matrix can be displayed also. The average residual and the average off-diagonal residual are also displayed. See the section “Assessment of Fit” on page 495 for more details.

• COSAN and LINEQS: Rank order of the largest normalized residuals if you specify the RESIDUAL, or at least the PRINT option.

• COSAN and LINEQS: Bar chart of the normalized residuals if you specify the RESIDUAL, or at least the PRINT option.

• COSAN and LINEQS: Value of the fit function \( F \). See the section “Estimation Criteria” on page 490 for more details. This output can be suppressed only by the NOPRINT option.

• COSAN and LINEQS: Goodness-of-fit index (GFI), adjusted goodness-of-fit index (AGFI), and root mean square residual (RMR) (Jöreskog and Sörbom 1985). See the section “Assessment of Fit” on page 495 for more details. This output can be suppressed only by the NOPRINT option.

• COSAN and LINEQS: Parsimonious goodness-of-fit index (PGFI) of Mulaik et al. (1989). See the section “Assessment of Fit” on page 495 for more detail. This output can be suppressed only by the NOPRINT option.

• COSAN and LINEQS: Overall \( \chi^2 \), \( df \), and Prob>Chi**2 if the METHOD= option is not ULS or DWLS. The \( \chi^2 \) measure is the optimum function value \( F \) multiplied by \((N - 1)\) if a CORR or COV matrix is analyzed or multiplied by \( N \) if a UCORR or UCOV matrix is analyzed; \( \chi^2 \) measures the likelihood ratio test statistic for the null hypothesis that the predicted matrix \( \mathbf{C} \) has the specified model structure against the alternative that \( \mathbf{C} \) is unconstrained. The notation Prob>Chi**2 means “the probability under the null hypothesis of obtaining a greater \( \chi^2 \) statistic than that observed.” This output can be suppressed only by the NOPRINT option.

• COSAN and LINEQS: If METHOD= is not ULS or DWLS, the value of the probability of close fit (Browne and Cudeck 1993).

SAS OnlineDoc™: Version 7-1
See the section “Assessment of Fit” on page 495 for more detail. This output can be suppressed only by the NOPRINT option.

- COSAN and LINEQS: If the value of the METHOD= option is not ULS or DWLS, the value of the Browne & Cudeck (1993) expected cross validation (ECVI) index and the lower and upper limits of the confidence interval. The size of the confidence interval is defined by the option ALPHAECV=α, 0 ≤ α ≤ 1. The default is α = 0.1, which corresponds to a 90% confidence interval. See the section “Assessment of Fit” on page 495 for more detail. This output can be suppressed only by the NOPRINT option.

- COSAN and LINEQS: If the value of the METHOD= option is not ULS or DWLS, Bentler’s (1989) Comparative Fit Index. See the section “Assessment of Fit” on page 495 for more detail. This output can be suppressed only by the NOPRINT option.

- COSAN and LINEQS: If you specify METHOD=ML or METHOD=GLS, the \( \chi^2 \) value and corresponding probability adjusted by the relative kurtosis coefficient \( r_2 \), which should be a close approximation of the \( \chi^2 \) value for elliptically distributed data (Browne 1982). See the section “Assessment of Fit” on page 495 for more detail. This output can be suppressed only by the NOPRINT option.

- COSAN and LINEQS: The Normal Theory Reweighted LS \( \chi^2 \) Value is displayed if METHOD= ML. Instead of the function value \( F_{ML} \), the reweighted goodness-of-fit function \( F_{GW,LS} \) is used. See the section “Assessment of Fit” on page 495 for more detail.

- COSAN and LINEQS: Akaike’s Information Criterion if the value of the METHOD= option is not ULS or DWLS. See the section “Assessment of Fit” on page 495. This output can be suppressed only by the NOPRINT option.

- COSAN and LINEQS: Bozdogan’s (1987) Consistent Information Criterion, CAIC. See the section “Assessment of Fit” on page 495. This output can be suppressed only by the NOPRINT option.

- COSAN and LINEQS: Schwarz’s Bayesian Criterion (SBC) if the value of the METHOD= option is not ULS or DWLS (Schwarz 1978). See the section “Assessment of Fit” on page 495. This output can be suppressed only by the NOPRINT option.

- COSAN and LINEQS: If the value of the METHOD= option is not ULS or DWLS, the following fit functions based on the overall \( \chi^2 \) value are displayed:
  - McDonald’s (McDonald and Hartmann 1992) measure of centrality
  - Parsimonious index of James, Mulaik, and Brett (1982)
  - Z-Test of Wilson and Hilferty (1931)
  - Bentler and Bonett’s (1980) nonnormed coefficient
  - Bentler and Bonett’s (1980) normed coefficient
  - Bollen’s (1986) normed index \( \rho_1 \)
  - Bollen’s (1989a) nonnormed index \( \Delta_2 \)
See the section “Assessment of Fit” on page 495 for more detail. This output can be suppressed only by the NOPRINT option.

- COSAN and LINEQS: Hoelter’s (1983) Critical N Index is displayed (Bollen 1989b, p. 277). See the section “Assessment of Fit” on page 495 for more detail. This output can be suppressed only by the NOPRINT option.

- COSAN and LINEQS: Equations of linear dependencies among the parameters used in the model specification if the information matrix is recognized as singular at the final solution.

- COSAN: Model matrices containing the parameter estimates. Except for ULS or DWLS estimates, the approximate standard errors and $t$ values are also displayed. This output is displayed if you specify the PESTIM option or at least the PSHORT option.

- LINEQS: Linear equations containing the parameter estimates. Except for ULS and DWLS estimates, the approximate standard errors and $t$ values are also displayed. This output is displayed if you specify the PESTIM option, or at least the PSHORT option.

- LINEQS: Variances and covariances of the exogenous variables. This output is displayed if you specify the PESTIM option, or at least the PSHORT.

- LINEQS: Linear equations containing the standardized parameter estimates. This output is displayed if you specify the PESTIM option, or at least the PSHORT option.

- LINEQS: Table of correlations among the exogenous variables. This output is displayed if you specify the PESTIM option, or at least the PSHORT option.

- LINEQS: Squared Multiple Correlations table, which displays the error variances of the endogenous variables. These are the diagonal elements of the predicted model matrix. Also displayed is the Total Variance and the $R^2$ values corresponding to all endogenous variables. See the section “Assessment of Fit” on page 495 for more detail. This output is displayed if you specify the PESTIM option, or at least the PSHORT option.

- LINEQS: If you specify the PDETERM or the PALL option, the total determination of all equations (DETAE), the total determination of the structural equations (DETSE), and the total determination of the manifest variables (DETMV) are displayed. See the section “Assessment of Fit” on page 495 for more details. If one of the determinants in the formulas is 0, the corresponding coefficient is displayed as a missing value. If there are structural equations, PROC CALIS also displays the Stability Coefficient of Reciprocal Causation, that is, the largest eigenvalue of the $BB'$ matrix, where $B$ is the causal coefficient matrix of the structural equations.

- LINEQS: The matrix of estimated covariances among the latent variables if you specify the PLATCOV option, or at least the PRINT option.

- LINEQS: The matrix of estimated covariances between latent and manifest variables used in the model if you specify the PLATCOV option, or at least the
PRINT option.

- **LINEQS**: The matrix \( \mathbf{FSR} \) of latent variable scores regression coefficients if you specify the PLATCOV option, or at least the PRINT option. The \( \mathbf{FSR} \) matrix is a generalization of Lawley and Maxwell’s (1971, p.109) factor scores regression matrix,

\[
\mathbf{FSR} = \mathbf{C}_{yx} \mathbf{C}_{xx}^{-1}
\]

where \( \mathbf{C}_{xx} \) is the \( n \times n \) predicted model matrix (predicted covariances among manifest variables) and \( \mathbf{C}_{yx} \) is the \( n_{lat} \times n \) matrix of the predicted covariances between latent and manifest variables. You can multiply the manifest observations by this matrix to estimate the scores of the latent variables used in your model.

- **LINEQS**: The matrix \( \mathbf{TEF} \) of total effects if you specify the TOTEFF option, or at least the PRINT option. For the LINEQS model, the matrix of total effects is

\[
\mathbf{TEF} = (\mathbf{I} - \mathbf{\beta})^{-1}\gamma - (\mathbf{I} : \theta)
\]

(For the LISREL model, refer to Jöreskog and Sörbom 1985). The matrix of indirect effects is displayed also.

- **FACTOR**: The matrix of rotated factor loadings and the orthogonal transformation matrix if you specify the ROTATE= and PESTIM options, or at least the PSHORT options.

- **FACTOR**: Standardized (rotated) factor loadings, variance estimates of endogenous variables, \( R^2 \) values, correlations among factors, and factor scores regression matrix, if you specify the PESTIM option, or at least the PSHORT option. The determination of manifest variables is displayed only if you specify the PDETERM option.

- **COSAN and LINEQS**: Univariate Lagrange multiplier and Wald test indices are displayed in matrix form if you specify the MODIFICATION (or MOD) or the PALL option. Those matrix locations that correspond to constants in the model in general contain three values: the value of the Lagrange multiplier, the corresponding probability (\( df = 1 \)), and the estimated change of the parameter value should the constant be changed to a parameter. If allowing the constant to be an estimated parameter would result in a singular information matrix, the string ‘sing’ is displayed instead of the Lagrange multiplier index. Those matrix locations that correspond to parameter estimates in the model contain the Wald test index and the name of the parameter in the model. See the section “Modification Indices” on page 519 for more detail.

- **COSAN and LINEQS**: Univariate Lagrange multiplier test indices for releasing equality constraints if you specify the MODIFICATION (or MOD) or the PALL option. See the section “Modification Indices” on page 519 for more detail.

- **COSAN and LINEQS**: Univariate Lagrange multiplier test indices for releasing active boundary constraints specified by the BOUNDS statement if you
specify the MODIFICATION (or MOD) or the PALL option. See the section “Modification Indices” on page 519 for more detail.

- COSAN and LINEQS: If the MODIFICATION (or MOD) or the PALL option is specified, the stepwise multivariate Wald test for constraining estimated parameters to zero constants is performed as long as the univariate probability is larger than the value specified in the PMW= option (default PMW=0.05). See the section “Modification Indices” on page 519 for more details.

**ODS Table Names**

PROC CALIS assigns a name to each table it creates. You can use these names to reference the table when using the Output Delivery System (ODS) to select tables and create output data sets. These names are listed in the following table. For more information on ODS, see Chapter 14, “Using the Output Delivery System.”

<table>
<thead>
<tr>
<th>ODS Table Name</th>
<th>Description</th>
<th>Statement</th>
<th>Option</th>
</tr>
</thead>
<tbody>
<tr>
<td>AddParms</td>
<td>Additional Parameters and Dependent Parameter</td>
<td>CALIS 1</td>
<td>(default), PINITIAL, PALL, PRINT</td>
</tr>
<tr>
<td>AveRes</td>
<td>Average Absolute/ Normalized/ Standardized/ Off-diagonal Residual</td>
<td>CALIS</td>
<td>RESIDUAL=, PALL, or PRINT</td>
</tr>
<tr>
<td>ContKurtosis</td>
<td>Contributions to Kurtosis</td>
<td>CALIS</td>
<td>KURTOSIS, SIMPLE, PALL, or PRINT</td>
</tr>
<tr>
<td>ConvergenceStatus</td>
<td>Convergence Status</td>
<td>CALIS</td>
<td>(default), PALL, PRINT, or PSHORT</td>
</tr>
<tr>
<td>Corr</td>
<td>Predicted or Sample Correlations</td>
<td>CALIS</td>
<td>PCORR, or PALL</td>
</tr>
<tr>
<td>CorrExogVar</td>
<td>Correlations Among Exogenous Variables</td>
<td>CALIS 2</td>
<td>(default), PESTIM, PALL, PRINT, or PSHORT</td>
</tr>
<tr>
<td>Cov</td>
<td>Predicted or Sample Covariances</td>
<td>CALIS</td>
<td>PCORR, PALL</td>
</tr>
<tr>
<td>CovExogVar</td>
<td>Covariances of Exogenous Variables</td>
<td>CALIS 2</td>
<td>(default), PESTIM, PALL, PRINT, or PSHORT</td>
</tr>
<tr>
<td>Determinant</td>
<td>Determinants</td>
<td>CALIS</td>
<td>(default)</td>
</tr>
<tr>
<td>Determination</td>
<td>Determination Values</td>
<td>CALIS 3</td>
<td>PDETERM</td>
</tr>
<tr>
<td>EstimatedParm</td>
<td>Estimated Parameter Matrices</td>
<td>CALIS</td>
<td>(default with the COSAN statement) (for LINEQS or RAM, use PRIVEC), PESTIM, PALL, PRINT, or PSHORT</td>
</tr>
<tr>
<td>FactScoreReg</td>
<td>Factor Score Regression Coefficients</td>
<td>CALIS 4</td>
<td>(default), PESTIM, PALL, PRINT</td>
</tr>
<tr>
<td>Fit</td>
<td>Fit</td>
<td>CALIS</td>
<td>(default), PALL, PRINT, PSHORT, or PSUMMARY</td>
</tr>
<tr>
<td>GenModInfo</td>
<td>General Modeling Information</td>
<td>CALIS</td>
<td>(default), SIMPLE, PALL, or PRINT</td>
</tr>
<tr>
<td>IndirectEffects</td>
<td>Indirect Effects of Exogenous on Endogenous</td>
<td>CALIS 2</td>
<td>TOTEFF, PALL, or PRINT</td>
</tr>
</tbody>
</table>
### Table 17.15. (continued)

<table>
<thead>
<tr>
<th>ODS Table Name</th>
<th>Description</th>
<th>Statement</th>
<th>Option</th>
</tr>
</thead>
<tbody>
<tr>
<td>Information</td>
<td>Information Matrix</td>
<td>CALIS</td>
<td>PCOVES (except for ULS or DWLS estimation)</td>
</tr>
<tr>
<td>InitialParm</td>
<td>Initial Parameter Matrices</td>
<td>CALIS</td>
<td>(default with the COSAN statement)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>(for LINEQS or RAM, use PRIVEC), PINITIAL, PALL, or PRINT</td>
</tr>
<tr>
<td>InputSym</td>
<td>Input Symmetric Matrix</td>
<td>CALIS</td>
<td>(default), PALL, or PRINT</td>
</tr>
<tr>
<td>IterHist</td>
<td>Iteration History</td>
<td>CALIS</td>
<td>(default), PALL, PRINT, or PSHORT</td>
</tr>
<tr>
<td>IterStart</td>
<td>Iteration Start</td>
<td>CALIS</td>
<td>(default), PALL, PRINT, or PSHORT</td>
</tr>
<tr>
<td>IterStop</td>
<td>Iteration Stop</td>
<td>CALIS</td>
<td>(default), PALL, PRINT, or PSHORT</td>
</tr>
<tr>
<td>Jacobian</td>
<td>Jacobi Column Pattern</td>
<td>CALIS</td>
<td>PJACPAT</td>
</tr>
<tr>
<td>Kurtosis</td>
<td>Kurtosis</td>
<td>CALIS</td>
<td>KURTOSIS, SIMPLE, PALL, or PRINT</td>
</tr>
<tr>
<td>LagrangeBoundary</td>
<td>Univariate Lagrange Multiplier Test, Boundary</td>
<td>CALIS</td>
<td>MODIFICATION or PALL</td>
</tr>
<tr>
<td>LatentScoreCoef</td>
<td>Latent Variable Coefficients</td>
<td>CALIS</td>
<td>PLATCOV, PALL, or PRINT</td>
</tr>
<tr>
<td>LinearEquations</td>
<td>Linear Equations</td>
<td>CALIS&lt;sup&gt;5&lt;/sup&gt;</td>
<td>(default), PESTIM with PRIMAT omitted, PALL, PRINT, or PSHORT</td>
</tr>
<tr>
<td>ModelStatement</td>
<td>Summary of Model</td>
<td>CALIS</td>
<td>(default), PALL, PRINT, or PSHORT</td>
</tr>
<tr>
<td>ModIndices</td>
<td>Lagrange Multiplier and Wald Test Indices</td>
<td>CALIS</td>
<td>MODIFICATION, or PALL with NOMOD omitted (except for ULS or DWLS estimation)</td>
</tr>
<tr>
<td>PredetElements</td>
<td>Predetermined Elements in the Predicted Moment Matrix</td>
<td>CALIS</td>
<td>PREDET, PALL, or when predetermined elements detected and PSUMMARY omitted</td>
</tr>
<tr>
<td>PredModel</td>
<td>Predicted Model Matrix</td>
<td>CALIS</td>
<td>PCORR or PALL</td>
</tr>
<tr>
<td>PredModelDet</td>
<td>Predicted Model Matrix Determinant</td>
<td>CALIS</td>
<td>PCORR or PALL</td>
</tr>
<tr>
<td>PredMomentLatent</td>
<td>Latent Variables Moments</td>
<td>CALIS&lt;sup&gt;2&lt;/sup&gt;</td>
<td>PLATCOV, PALL, or PRINT</td>
</tr>
<tr>
<td>PredMomentManLat</td>
<td>Manifest and Latent Variables Moments</td>
<td>CALIS</td>
<td>PLATCOV, PALL, or PRINT</td>
</tr>
<tr>
<td>ProblemDescription</td>
<td>Problem Description</td>
<td>CALIS</td>
<td>(default), PALL, PRINT, or PSHORT</td>
</tr>
<tr>
<td>RAMEstimates</td>
<td>RAM Pattern and Values</td>
<td>CALIS&lt;sup&gt;6&lt;/sup&gt;</td>
<td>(default), PESTIM, PALL, PRINT, or PSHORT</td>
</tr>
<tr>
<td>RAMStandEst</td>
<td>Standardized Coefficients</td>
<td>CALIS&lt;sup&gt;6&lt;/sup&gt;</td>
<td>(default), PESTIM, PALL, PRINT, or PSHORT</td>
</tr>
<tr>
<td>RankOrdLagrange</td>
<td>Rank Order of Largest Lagrange Multipliers</td>
<td>CALIS</td>
<td>MODIFICATION or PALL</td>
</tr>
<tr>
<td>RankOrdRes</td>
<td>Rank Order of Largest Residuals</td>
<td>CALIS</td>
<td>(default), RESIDUAL, PALL, or PRINT</td>
</tr>
<tr>
<td>Residuals</td>
<td>Residual Matrix</td>
<td>CALIS</td>
<td>(default), RESIDUAL, PALL, or PRINT</td>
</tr>
<tr>
<td>ResidualsDist</td>
<td>Distribution of Residuals</td>
<td>CALIS</td>
<td>(default), RESIDUAL, PALL, or PRINT</td>
</tr>
</tbody>
</table>
Table 17.15. (continued)

<table>
<thead>
<tr>
<th>ODS Table Name</th>
<th>Description</th>
<th>Statement Comment</th>
<th>Option Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>Rotated</td>
<td>Rotated Factor Loadings</td>
<td>CALIS 1</td>
<td>(default), PESTIM, or ROTATE= with PSUMMARY omitted</td>
</tr>
<tr>
<td>SetCovExogVar</td>
<td>Set Covariances of Exogenous Manifest Variables</td>
<td>CALIS 2</td>
<td>(default), PINICIAL, PALL, or PRINT</td>
</tr>
<tr>
<td>SimpleStatistics</td>
<td>Simple Statistics</td>
<td>CALIS</td>
<td>(default), SIMPLE, PALL, or PRINT</td>
</tr>
<tr>
<td>SqMultCorr</td>
<td>Squared Multiple Correlations</td>
<td>CALIS 2</td>
<td>(default), PESTIM, PALL, PRINT, or PSHORT</td>
</tr>
<tr>
<td>Stability</td>
<td>Stability of Reciprocal Causation</td>
<td>CALIS 2</td>
<td>PDETERM</td>
</tr>
<tr>
<td>StdLoadings</td>
<td>Standardized Factor Loadings</td>
<td>CALIS 4</td>
<td>(default), PESTIM, or ROTATE= with PSUMMARY omitted</td>
</tr>
<tr>
<td>StructEq</td>
<td>Structural Equations Corresponding to the Dependent Variables</td>
<td>CALIS 2</td>
<td>PDETERM</td>
</tr>
<tr>
<td>SumSqDif</td>
<td>Sum of Squared Differences for Predetermined Elements</td>
<td>CALIS</td>
<td>PREDET, PALL, or when predetermined elements detected in the model and PSUMMARY omitted</td>
</tr>
<tr>
<td>Symmetric</td>
<td>Input Symmetric Matrix</td>
<td>CALIS</td>
<td>PCORR or PALL</td>
</tr>
<tr>
<td>TotalEffects</td>
<td>Total Effects of Exogenous on Endogenous Variables</td>
<td>CALIS 2</td>
<td>TOTEFF, PALL, or PRINT</td>
</tr>
<tr>
<td>Transformation</td>
<td>Orthogonal Transformation Matrix</td>
<td>CALIS 4</td>
<td>(default), PESTIM, or ROTATE= with PSUMMARY omitted</td>
</tr>
<tr>
<td>VarExogVar</td>
<td>Variances of Exogenous Variables</td>
<td>CALIS 2</td>
<td>(default), PESTIM, PALL, PRINT, or PSHORT</td>
</tr>
<tr>
<td>Variables</td>
<td>Variables and Their Types</td>
<td>CALIS 2</td>
<td>(default), PALL, PRINT, or PSHORT</td>
</tr>
<tr>
<td>VarSel</td>
<td>Automatic Variable Selection</td>
<td>CALIS 2</td>
<td>(default)</td>
</tr>
<tr>
<td>VectorOutput</td>
<td>Vector Output</td>
<td>CALIS</td>
<td>(default with the COSAN statement), PRIVEC</td>
</tr>
<tr>
<td>WaldTest</td>
<td>Stepwise Multivariate Wald Test</td>
<td>CALIS</td>
<td>MODIFICATION or PALL</td>
</tr>
<tr>
<td>WeightsDet</td>
<td>Determinant of Weights</td>
<td>CALIS</td>
<td>PWEIGHT, or PALL (for GLS, WLS, and DWLS estimation)</td>
</tr>
<tr>
<td>Weights</td>
<td>Weight Matrix</td>
<td>CALIS</td>
<td>PWEIGHT, or PALL (for GLS, WLS, and DWLS estimation)</td>
</tr>
</tbody>
</table>

1. with the PARAMETERS statement or when program statements are used
2. with the LINEQS or RAM statement
3. with the FACTOR, LINEQS, or RAM statement
4. with the FACTOR statement only
5. with the LINEQS statement only
6. with the RAM statement only
Examples

Example 17.1. Path Analysis: Stability of Alienation

The following covariance matrix from Wheaton, Muthen, Alwin, and Summers (1977) has served to illustrate the performance of several implementations for the analysis of structural equation models. Two different models have been analyzed by an early implementation of LISREL and are mentioned in Jöreskog (1978). You also can find a more detailed discussion of these models in the LISREL VI manual (Jöreskog and Sörbom 1985). A slightly modified model for this covariance matrix is included in the EQS 2.0 manual (Bentler 1985, p. 28). The path diagram of this model is displayed in Figure 17.1. The same model is reanalyzed here by PROC CALIS. However, for the analysis with the EQS implementation, the last variable (V6) is rescaled by a factor of 0.1 to make the matrix less ill-conditioned. Since the Levenberg-Marquardt or Newton-Raphson optimization techniques are used with PROC CALIS, rescaling the data matrix is not necessary and, therefore, is not done here. The results reported here reflect the estimates based on the original covariance matrix.

```
DATA Wheaton(TYPE=COV);
TITLE "Stability of Alienation";
TITLE2 "Data Matrix of WHEATON, MUTHEN, ALWIN & SUMMERS (1977)";
_TYPE_ = 'COV'; INPUT _NAME_ $ V1-V6;
LABEL V1='Anomia (1967)' V2='Anomia (1971)' V3='Education'
     V4='Powerlessness (1967)' V5='Powerlessness (1971)'
     V6='Occupational Status Index';
datalines;
V1 11.834 . . . . . 
;
proc calis cov data=Wheaton tech=nr edf=931 pall;
lineqs
  V1 = F1 + E1,
  V2 = .833 F1 + E2,
  V3 = F2 + E3,
  V4 = .833 F2 + E4,
  V5 = F3 + E5,
  V6 = Lamb (.5) F3 + E6,
  F1 = Gam1(-.5) F3 + D1,
  F2 = Beta (.5) F1 + Gam2(-.5) F3 + D2;
std
  E1-E6 = Thel-The2 Thel-The4 (6 * 3.),
  D1-D2 = Psi1-Psi2 (2 * 4.),
  F3 = Phi (6.);
```

SAS OnlineDoc™: Version 7-1
Example 17.1. Path Analysis: Stability of Alienation

Cov
E1 E3 = The5 (.2),
E4 E2 = The5 (.2);
RUN;

The COV option in the PROC CALIS statement requests the analysis of the covariance matrix. Without the COV option, the correlation matrix would be computed and analyzed. Since no METHOD= option has been used, maximum likelihood estimates are computed by default. The TECH=NR option requests the Newton-Raphson optimization method. The PALL option produces the almost complete set of displayed output, as displayed in Output 17.1.1 through Output 17.1.11. Note that, when you specify the PALL option, you can produce large amounts of output. The PALL option is used in this example to show how you can get a wide spectrum of useful information from PROC CALIS.

Output 17.1.1 displays the model specification in matrix terms, followed by the lists of endogenous and exogenous variables. Equations and initial parameter estimates are also displayed. You can use this information to ensure that the desired model is the model being analyzed.

Output 17.1.1. Model Specification and Initial Values

<table>
<thead>
<tr>
<th>Term</th>
<th>Matrix</th>
<th>Rows</th>
<th>Columns</th>
<th>Matrix Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td><em>SEL</em></td>
<td>6</td>
<td>17</td>
<td>SELECTION</td>
</tr>
<tr>
<td>2</td>
<td><em>BETA</em></td>
<td>17</td>
<td>17</td>
<td>EQSBETA</td>
</tr>
<tr>
<td>3</td>
<td><em>GAMMA</em></td>
<td>17</td>
<td>9</td>
<td>EQSGAMMA</td>
</tr>
<tr>
<td>4</td>
<td><em>PHI</em></td>
<td>9</td>
<td>9</td>
<td>SYMMETRIC</td>
</tr>
</tbody>
</table>

The 8 Endogenous Variables

- Manifest: V1, V2, V3, V4, V5, V6
- Latent: F1, F2

The 9 Exogenous Variables

- Manifest: F3
- Latent: Error: E1, E2, E3, E4, E5, E6, D1, D2
Output 17.1.1. (continued)

<table>
<thead>
<tr>
<th>Stability of Alienation</th>
<th>Data Matrix of WHEATON, MUTHEN, ALWIN &amp; SUMMERS (1977)</th>
</tr>
</thead>
<tbody>
<tr>
<td>The CALIS Procedure</td>
<td>Covariance Structure Analysis: Pattern and Initial Values</td>
</tr>
<tr>
<td>V1 = 1.0000 F1 + 1.0000 E1</td>
<td></td>
</tr>
<tr>
<td>V2 = 0.8330 F1 + 1.0000 E2</td>
<td></td>
</tr>
<tr>
<td>V3 = 1.0000 F2 + 1.0000 E3</td>
<td></td>
</tr>
<tr>
<td>V4 = 0.8330 F2 + 1.0000 E4</td>
<td></td>
</tr>
<tr>
<td>V5 = 1.0000 F3 + 1.0000 E5</td>
<td></td>
</tr>
<tr>
<td>V6 = 0.5000*F3 + 1.0000 E6</td>
<td></td>
</tr>
</tbody>
</table>

Lamb

Output 17.1.1. (continued)

<table>
<thead>
<tr>
<th>Stability of Alienation</th>
<th>Data Matrix of WHEATON, MUTHEN, ALWIN &amp; SUMMERS (1977)</th>
</tr>
</thead>
<tbody>
<tr>
<td>The CALIS Procedure</td>
<td>Covariance Structure Analysis: Pattern and Initial Values</td>
</tr>
<tr>
<td>F1 = -0.5000*F3 + 1.0000 D1</td>
<td></td>
</tr>
<tr>
<td>F2 = 0.5000<em>F1 + -0.5000</em>F3 + 1.0000 D2</td>
<td></td>
</tr>
</tbody>
</table>

Gam1

Beta Gam2

Output 17.1.1. (continued)

<table>
<thead>
<tr>
<th>Stability of Alienation</th>
<th>Data Matrix of WHEATON, MUTHEN, ALWIN &amp; SUMMERS (1977)</th>
</tr>
</thead>
<tbody>
<tr>
<td>The CALIS Procedure</td>
<td>Covariance Structure Analysis: Pattern and Initial Values</td>
</tr>
<tr>
<td>Variances of Exogenous Variables</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Variable Parameter</th>
<th>Estimate</th>
</tr>
</thead>
<tbody>
<tr>
<td>F3 Phi</td>
<td>6.0000</td>
</tr>
<tr>
<td>E1 The1</td>
<td>3.0000</td>
</tr>
<tr>
<td>E2 The2</td>
<td>3.0000</td>
</tr>
<tr>
<td>E3 The1</td>
<td>3.00000</td>
</tr>
<tr>
<td>E4 The2</td>
<td>3.0000</td>
</tr>
<tr>
<td>E5 The3</td>
<td>3.0000</td>
</tr>
<tr>
<td>E6 The4</td>
<td>3.00000</td>
</tr>
<tr>
<td>D1 Psi1</td>
<td>4.00000</td>
</tr>
<tr>
<td>D2 Psi2</td>
<td>4.00000</td>
</tr>
</tbody>
</table>

Covariances Among Exogenous Variables

<table>
<thead>
<tr>
<th>Var1</th>
<th>Var2</th>
<th>Parameter</th>
<th>Estimate</th>
</tr>
</thead>
<tbody>
<tr>
<td>E1</td>
<td>E3</td>
<td>The5</td>
<td>0.20000</td>
</tr>
<tr>
<td>E2</td>
<td>E4</td>
<td>The5</td>
<td>0.20000</td>
</tr>
</tbody>
</table>

General modeling information and simple descriptive statistics are displayed in Output 17.1.2. Because the input data set contains only the covariance matrix, the means
of the manifest variables are assumed to be zero. Note that this has no impact on the estimation, unless a mean structure model is being analyzed. The twelve parameter estimates in the model and their respective locations in the parameter matrices are also displayed. Each of the parameters, $\theta_1$, $\theta_2$, and $\theta_5$, are common to two elements in the parameter matrix $\varphi$.

**Output 17.1.2.** Simple Statistics and Parameter Locations

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Mean</th>
<th>Std Dev</th>
</tr>
</thead>
<tbody>
<tr>
<td>V1 Anomia (1967)</td>
<td>0</td>
<td>3.44005814</td>
</tr>
<tr>
<td>V2 Anomia (1971)</td>
<td>0</td>
<td>3.06006536</td>
</tr>
<tr>
<td>V3 Education</td>
<td>0</td>
<td>3.54000565</td>
</tr>
<tr>
<td>V4 Powerlessness (1967)</td>
<td>0</td>
<td>3.16006329</td>
</tr>
<tr>
<td>V5 Powerlessness (1971)</td>
<td>0</td>
<td>3.10000000</td>
</tr>
<tr>
<td>V6 Occupational Status Index</td>
<td>0</td>
<td>21.21999057</td>
</tr>
</tbody>
</table>

**Output 17.1.2.** (continued)

<table>
<thead>
<tr>
<th>Covariances</th>
<th>V1</th>
<th>V2</th>
<th>V3</th>
<th>V4</th>
<th>V5</th>
<th>V6</th>
</tr>
</thead>
</table>

Determinant 6080570  Ln 15.620609
Output 17.1.2. (continued)

Stability of Alienation
Data Matrix of WHEATON, MUTHEN, ALWIN & SUMMERS (1977)

The CALIS Procedure
Covariance Structure Analysis: Maximum Likelihood Estimation

Vector of Initial Estimates

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Estimate</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 Beta</td>
<td>0.50000</td>
<td>Matrix Entry: <em>BETA</em>[8:7]</td>
</tr>
<tr>
<td>2 Lamb</td>
<td>0.50000</td>
<td>Matrix Entry: <em>GAMMA</em>[6:1]</td>
</tr>
<tr>
<td>3 Gam1</td>
<td>-0.50000</td>
<td>Matrix Entry: <em>GAMMA</em>[7:1]</td>
</tr>
<tr>
<td>4 Gam2</td>
<td>-0.50000</td>
<td>Matrix Entry: <em>GAMMA</em>[8:1]</td>
</tr>
<tr>
<td>5 Phi</td>
<td>6.00000</td>
<td>Matrix Entry: <em>PHI</em>[1:1]</td>
</tr>
<tr>
<td>7 The2</td>
<td>3.00000</td>
<td>Matrix Entry: <em>PHI</em>[3:3] <em>PHI</em>[5:5]</td>
</tr>
<tr>
<td>8 The5</td>
<td>0.20000</td>
<td>Matrix Entry: <em>PHI</em>[4:2] <em>PHI</em>[5:3]</td>
</tr>
<tr>
<td>9 The3</td>
<td>3.00000</td>
<td>Matrix Entry: <em>PHI</em>[6:6]</td>
</tr>
<tr>
<td>10 The4</td>
<td>3.00000</td>
<td>Matrix Entry: <em>PHI</em>[7:7]</td>
</tr>
<tr>
<td>11 Psi1</td>
<td>4.00000</td>
<td>Matrix Entry: <em>PHI</em>[8:8]</td>
</tr>
<tr>
<td>12 Psi2</td>
<td>4.00000</td>
<td>Matrix Entry: <em>PHI</em>[9:9]</td>
</tr>
</tbody>
</table>

PROC CALIS examines whether each element in the moment matrix is modeled by the parameters defined in the model. If an element is not structured by the model parameters, it is predetermined by its observed value. This occurs, for example, when there are exogenous manifest variables in the model. If present, the predetermined values of the elements will be displayed. In the current example, the ‘.’ displayed for all elements in the predicted moment matrix (Output 17.1.3) indicates that there are no predetermined elements in the model.

Output 17.1.3. Predetermined Elements

Stability of Alienation
Data Matrix of WHEATON, MUTHEN, ALWIN & SUMMERS (1977)

The CALIS Procedure
Covariance Structure Analysis: Maximum Likelihood Estimation

Predetermined Elements of the Predicted Moment Matrix

<table>
<thead>
<tr>
<th>V1</th>
<th>V2</th>
<th>V3</th>
<th>V4</th>
<th>V5</th>
<th>V6</th>
</tr>
</thead>
<tbody>
<tr>
<td>V1 Anomia (1967)</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
</tr>
<tr>
<td>V2 Anomia (1971)</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
</tr>
<tr>
<td>V3 Education</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
</tr>
<tr>
<td>V4 Powerlessness (1967)</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td></td>
</tr>
<tr>
<td>V5 Powerlessness (1971)</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td></td>
</tr>
<tr>
<td>V6 Occupational Status Index</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td></td>
</tr>
</tbody>
</table>

Sum of Squared Differences 0

Output 17.1.4 displays the optimization information. You can check this table to determine whether the convergence criterion is satisfied. PROC CALIS displays an error message when problematic solutions are encountered.
Output 17.1.4. Optimization

Stability of Alienation
Data Matrix of WHEATON, MUTHEN, ALWIN & SUMMERS (1977)

The CALIS Procedure
Covariance Structure Analysis: Maximum Likelihood Estimation

Parameter Estimates 12
Functions (Observations) 21

Optimization Start
Active Constraints 0 Objective Function 119.33282242
Max Abs Gradient Element 74.016932345

<table>
<thead>
<tr>
<th>Iter</th>
<th>Restarts</th>
<th>Function Calls</th>
<th>Active Constraints</th>
<th>Objective Function</th>
<th>Max Abs Gradient Element</th>
<th>Ridge</th>
<th>Predicted Change</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0</td>
<td>2</td>
<td>0</td>
<td>0.82689</td>
<td>118.5</td>
<td>1.3507</td>
<td>0.0154</td>
</tr>
<tr>
<td>2</td>
<td>0</td>
<td>3</td>
<td>0</td>
<td>0.09859</td>
<td>0.7283</td>
<td>0.2330</td>
<td>0.0716</td>
</tr>
<tr>
<td>3</td>
<td>0</td>
<td>4</td>
<td>0</td>
<td>0.01581</td>
<td>0.0828</td>
<td>0.00684</td>
<td>0.0185</td>
</tr>
<tr>
<td>4</td>
<td>0</td>
<td>5</td>
<td>0</td>
<td>0.01449</td>
<td>0.00132</td>
<td>0.000286</td>
<td>0.0142</td>
</tr>
<tr>
<td>5</td>
<td>0</td>
<td>6</td>
<td>0</td>
<td>0.01448</td>
<td>9.936E-7</td>
<td>0.000045</td>
<td>0.0153</td>
</tr>
<tr>
<td>6</td>
<td>0</td>
<td>7</td>
<td>0</td>
<td>0.01448</td>
<td>4.227E-9</td>
<td>1.685E-6</td>
<td>0.0156</td>
</tr>
</tbody>
</table>

Optimization Results
Iterations 6 Function Calls 8
Jacobian Calls 7 Active Constraints 0
Objective Function 0.0144844811 Max Abs Gradient Element 1.6847829E-6
Ridge 0 Actual Over Pred Change 1.0563187228

ABSGCONV convergence criterion satisfied.

The predicted model matrix is displayed next, followed by a list of model test statistics or fit indices (Output 17.1.5). Depending on your modeling philosophy, some indices may be preferred to others. In this example, all indices and test statistics point to a good fit of the model.

Output 17.1.5. Predicted Model Matrix and Fit Statistics

Stability of Alienation
Data Matrix of WHEATON, MUTHEN, ALWIN & SUMMERS (1977)

The CALIS Procedure
Covariance Structure Analysis: Maximum Likelihood Estimation

Predicted Model Matrix

<table>
<thead>
<tr>
<th></th>
<th>V1</th>
<th>V2</th>
<th>V3</th>
<th>V4</th>
<th>V5</th>
<th>V6</th>
</tr>
</thead>
</table>

Determinant 6169285 Ln 15.635094
### Output 17.1.5. (continued)

#### Stability of Alienation
Data Matrix of WHEATON, MUTHEN, ALWIN & SUMMERS (1977)

The CALIS Procedure
Covariance Structure Analysis: Maximum Likelihood Estimation

<table>
<thead>
<tr>
<th>Statistic</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fit Function</td>
<td>0.0145</td>
</tr>
<tr>
<td>Goodness of Fit Index (GFI)</td>
<td>0.9953</td>
</tr>
<tr>
<td>GFI Adjusted for Degrees of Freedom (AGFI)</td>
<td>0.9890</td>
</tr>
<tr>
<td>Root Mean Square Residual (RMR)</td>
<td>0.2281</td>
</tr>
<tr>
<td>Parsimonious GFI (Mulaik, 1989)</td>
<td>0.5972</td>
</tr>
<tr>
<td>Chi-Square</td>
<td>13.4851</td>
</tr>
<tr>
<td>Chi-Square DF</td>
<td>9</td>
</tr>
<tr>
<td>Pr &gt; Chi-Square</td>
<td>0.1419</td>
</tr>
<tr>
<td>Independence Model Chi-Square</td>
<td>2131.4</td>
</tr>
<tr>
<td>Independence Model Chi-Square DF</td>
<td>15</td>
</tr>
<tr>
<td>RMSEA Estimate</td>
<td>0.0231</td>
</tr>
<tr>
<td>RMSEA 90% Lower Confidence Limit</td>
<td></td>
</tr>
<tr>
<td>RMSEA 90% Upper Confidence Limit</td>
<td>0.0470</td>
</tr>
<tr>
<td>ECVI Estimate</td>
<td>0.0405</td>
</tr>
<tr>
<td>ECVI 90% Lower Confidence Limit</td>
<td></td>
</tr>
<tr>
<td>ECVI 90% Upper Confidence Limit</td>
<td>0.0556</td>
</tr>
<tr>
<td>Probability of Close Fit</td>
<td>0.9705</td>
</tr>
<tr>
<td>Bentler’s Comparative Fit Index</td>
<td>0.9979</td>
</tr>
<tr>
<td>Normal Theory Reweighted LS Chi-Square</td>
<td>13.2804</td>
</tr>
<tr>
<td>Akaike’s Information Criterion</td>
<td>-4.5149</td>
</tr>
<tr>
<td>Bozdogan’s (1987) CAIC</td>
<td>-57.0509</td>
</tr>
<tr>
<td>Schwarz’s Bayesian Criterion</td>
<td>-48.0509</td>
</tr>
<tr>
<td>McDonald’s (1989) Centrality</td>
<td>0.9976</td>
</tr>
<tr>
<td>Bentler &amp; Bonett’s (1980) Non-normed Index</td>
<td>0.9965</td>
</tr>
<tr>
<td>Bentler &amp; Bonett’s (1980) NFI</td>
<td>0.9937</td>
</tr>
<tr>
<td>James, Mulaik, &amp; Brett (1982) Parsimonious NFI</td>
<td>0.5962</td>
</tr>
<tr>
<td>Z-Test of Wilson &amp; Hilferty (1931)</td>
<td>1.0754</td>
</tr>
<tr>
<td>Bollen (1986) Normed Index Rho1</td>
<td>0.9895</td>
</tr>
<tr>
<td>Bollen (1988) Non-normed Index Delta2</td>
<td>0.9979</td>
</tr>
<tr>
<td>Hoelter’s (1983) Critical N</td>
<td>1170</td>
</tr>
</tbody>
</table>

PROC CALIS can perform a detailed residual analysis. Large residuals may indicate misspecification of the model. In Output 17.1.6 for example, note the table for the 10 largest asymptotically standardized residuals. As the table shows, the specified model performs the poorest concerning the variable V5 and its covariance with V2, V1, and V3. This may be the result of a misspecification of the model equation for V5. However, because the model fit is quite good, such a possible misspecification may have no practical significance and is not a serious concern in the analysis.
Output 17.1.6. Residual Analysis

Stability of Alienation
Data Matrix of WHEATON, MUTHEN, ALWIN & SUMMERS (1977)

The CALIS Procedure
Covariance Structure Analysis: Maximum Likelihood Estimation

Residual Matrix

<table>
<thead>
<tr>
<th></th>
<th>V1</th>
<th>V2</th>
<th>V3</th>
<th>V4</th>
<th>V5</th>
<th>V6</th>
</tr>
</thead>
<tbody>
<tr>
<td>V1 Anomia (1967)</td>
<td>-.0699063150</td>
<td>0.0364095216</td>
<td>-.0111621061</td>
<td>-.1519958205</td>
<td>0.3289115712</td>
<td>0.4778815840</td>
</tr>
<tr>
<td>V2 Anomia (1971)</td>
<td>0.0364095216</td>
<td>0.1254934661</td>
<td>0.1560041795</td>
<td>0.0113511059</td>
<td>-.4171296612</td>
<td>-.1910576405</td>
</tr>
<tr>
<td>V3 Education</td>
<td>-.0111621061</td>
<td>0.1560041795</td>
<td>-.0837499788</td>
<td>-.0085562504</td>
<td>0.1406088766</td>
<td>-.2383085022</td>
</tr>
<tr>
<td>V4 Powerlessness (1967)</td>
<td>-.1519958205</td>
<td>0.0113511059</td>
<td>-.0085562504</td>
<td>0.1406088766</td>
<td>-.2383085022</td>
<td>-.5923698474</td>
</tr>
<tr>
<td>V5 Powerlessness (1971)</td>
<td>0.3289115712</td>
<td>-.4171296612</td>
<td>0.2246560598</td>
<td>-.2383085022</td>
<td>0.0000000000</td>
<td>0.0000000000</td>
</tr>
<tr>
<td>V6 Occupational Status Index</td>
<td>0.4778815840</td>
<td>-.1910576405</td>
<td>0.0798873380</td>
<td>-.5923698474</td>
<td>0.0000000000</td>
<td>0.0000000000</td>
</tr>
</tbody>
</table>

Average Absolute Residual 0.153928
Average Off-diagonal Absolute Residual 0.195045

Output 17.1.6. (continued)

Stability of Alienation
Data Matrix of WHEATON, MUTHEN, ALWIN & SUMMERS (1977)

The CALIS Procedure
Covariance Structure Analysis: Maximum Likelihood Estimation

Rank Order of the 10 Largest Residuals

<table>
<thead>
<tr>
<th>Row</th>
<th>Column</th>
<th>Residual</th>
</tr>
</thead>
<tbody>
<tr>
<td>V6</td>
<td>V4</td>
<td>-.59237</td>
</tr>
<tr>
<td>V6</td>
<td>V1</td>
<td>0.47788</td>
</tr>
<tr>
<td>V5</td>
<td>V2</td>
<td>-.41713</td>
</tr>
<tr>
<td>V5</td>
<td>V1</td>
<td>0.32891</td>
</tr>
<tr>
<td>V5</td>
<td>V4</td>
<td>-.23831</td>
</tr>
<tr>
<td>V5</td>
<td>V3</td>
<td>0.22466</td>
</tr>
<tr>
<td>V6</td>
<td>V2</td>
<td>-.19106</td>
</tr>
<tr>
<td>V3</td>
<td>V1</td>
<td>0.15600</td>
</tr>
<tr>
<td>V4</td>
<td>V1</td>
<td>-.15200</td>
</tr>
<tr>
<td>V4</td>
<td>V4</td>
<td>0.14061</td>
</tr>
</tbody>
</table>

Output 17.1.6. (continued)

Stability of Alienation
Data Matrix of WHEATON, MUTHEN, ALWIN & SUMMERS (1977)

The CALIS Procedure
Covariance Structure Analysis: Maximum Likelihood Estimation

Asymptotically Standardized Residual Matrix

<table>
<thead>
<tr>
<th></th>
<th>V1</th>
<th>V2</th>
<th>V3</th>
<th>V4</th>
<th>V5</th>
<th>V6</th>
</tr>
</thead>
<tbody>
<tr>
<td>V1 Anomia (1967)</td>
<td>-.3085487887</td>
<td>0.5266544524</td>
<td>-.0561888626</td>
<td>-.865070455</td>
<td>2.553366366</td>
<td>0.464666661</td>
</tr>
<tr>
<td>V2 Anomia (1971)</td>
<td>0.5266544524</td>
<td>0.0543634844</td>
<td>0.876210855</td>
<td>0.057354415</td>
<td>-.763708659</td>
<td>-.170127806</td>
</tr>
<tr>
<td>V3 Education</td>
<td>-.0561888626</td>
<td>0.876210855</td>
<td>-.354347092</td>
<td>-.121874301</td>
<td>1.697931678</td>
<td>0.070202664</td>
</tr>
<tr>
<td>V4 Powerlessness (1967)</td>
<td>-.865070455</td>
<td>0.057354415</td>
<td>-.121874301</td>
<td>0.584930625</td>
<td>-1.557412695</td>
<td>-.495902427</td>
</tr>
<tr>
<td>V5 Powerlessness (1971)</td>
<td>2.553366366</td>
<td>-2.763708659</td>
<td>1.697931678</td>
<td>1.557412695</td>
<td>0.000000000</td>
<td>0.000000000</td>
</tr>
<tr>
<td>V6 Occupational Status Index</td>
<td>0.464666661</td>
<td>-.170127806</td>
<td>0.070202664</td>
<td>-.495902427</td>
<td>0.000000000</td>
<td>0.000000000</td>
</tr>
</tbody>
</table>

Average Standardized Residual 0.646622
Average Off-diagonal Standardized Residual 0.818457
Output 17.1.6. (continued)

Stability of Alienation
Data Matrix of WHEATON, MUTHEN, ALWIN & SUMMERS (1977)

The CALIS Procedure
Covariance Structure Analysis: Maximum Likelihood Estimation

Rank Order of the 10 Largest Asymptotically Standardized Residuals

<table>
<thead>
<tr>
<th>Row</th>
<th>Column</th>
<th>Residual</th>
</tr>
</thead>
<tbody>
<tr>
<td>V5</td>
<td>V2</td>
<td>-2.76371</td>
</tr>
<tr>
<td>V5</td>
<td>V1</td>
<td>2.55337</td>
</tr>
<tr>
<td>V5</td>
<td>V3</td>
<td>1.69793</td>
</tr>
<tr>
<td>V5</td>
<td>V4</td>
<td>-1.55741</td>
</tr>
<tr>
<td>V3</td>
<td>V2</td>
<td>0.87612</td>
</tr>
<tr>
<td>V4</td>
<td>V1</td>
<td>-0.86507</td>
</tr>
<tr>
<td>V4</td>
<td>V4</td>
<td>0.58493</td>
</tr>
<tr>
<td>V2</td>
<td>V1</td>
<td>0.52665</td>
</tr>
<tr>
<td>V6</td>
<td>V4</td>
<td>-0.49598</td>
</tr>
<tr>
<td>V6</td>
<td>V1</td>
<td>0.46487</td>
</tr>
</tbody>
</table>

Distribution of Asymptotically Standardized Residuals

Each * Represents 1 Residuals

<table>
<thead>
<tr>
<th>-------Range-------</th>
<th>Freq</th>
<th>Percent</th>
</tr>
</thead>
<tbody>
<tr>
<td>-3.00000 -2.75000</td>
<td>1</td>
<td>4.76</td>
</tr>
<tr>
<td>-2.75000 -2.50000</td>
<td>0</td>
<td>0.00</td>
</tr>
<tr>
<td>-2.50000 -2.25000</td>
<td>0</td>
<td>0.00</td>
</tr>
<tr>
<td>-2.25000 -2.00000</td>
<td>0</td>
<td>0.00</td>
</tr>
<tr>
<td>-2.00000 -1.75000</td>
<td>0</td>
<td>0.00</td>
</tr>
<tr>
<td>-1.75000 -1.50000</td>
<td>1</td>
<td>4.76</td>
</tr>
<tr>
<td>-1.50000 -1.25000</td>
<td>0</td>
<td>0.00</td>
</tr>
<tr>
<td>-1.25000 -1.00000</td>
<td>0</td>
<td>0.00</td>
</tr>
<tr>
<td>-1.00000 -0.75000</td>
<td>1</td>
<td>4.76</td>
</tr>
<tr>
<td>-0.75000 -0.50000</td>
<td>0</td>
<td>0.00</td>
</tr>
</tbody>
</table>
| -0.50000 -0.25000    | 3    | 14.29   ***
| -0.25000 0           | 3    | 14.29   ***
| 0 0.25000           | 6    | 28.57   ******
| 0.25000 0.50000      | 1    | 4.76    |
| 0.50000 0.75000      | 2    | 9.52    **
| 0.75000 1.00000      | 1    | 4.76    |
| 1.00000 1.25000      | 0    | 0.00    |
| 1.25000 1.50000      | 0    | 0.00    |
| 1.50000 1.75000      | 1    | 4.76    |
| 1.75000 2.00000      | 0    | 0.00    |
| 2.00000 2.25000      | 0    | 0.00    |
| 2.25000 2.50000      | 0    | 0.00    |
| 2.50000 2.75000      | 1    | 4.76    |
Output 17.1.7 displays the equations and parameter estimates. Each parameter estimate is displayed with its standard error and the corresponding $t$ ratio. As a general rule, a $t$ ratio larger than 2 represents a statistically significant departure from 0. From these results, it is observed that both $F_1$ (Alienation 1967) and $F_2$ (Alienation 1971) are regressed negatively on $F_3$ (Socioeconomic Status), and $F_1$ has a positive effect on $F_2$. The estimates and significance tests for the variance and covariance of the exogenous variables are also displayed.

**Output 17.1.7. Equations and Parameter Estimates**

\[
\begin{align*}
V_1 &= 1.0000 \ F_1 + 1.0000 \ E_1 \\
V_2 &= 0.8330 \ F_1 + 1.0000 \ E_2 \\
V_3 &= 1.0000 \ F_2 + 1.0000 \ E_3 \\
V_4 &= 0.8330 \ F_2 + 1.0000 \ E_4 \\
V_5 &= 1.0000 \ F_3 + 1.0000 \ E_5 \\
V_6 &= 5.3688*F_3 + 1.0000 \ E_6 \\
\end{align*}
\]

\[
\begin{align*}
\text{Std Err} &= 0.4337 \ \text{Lamb} \\
t \ \text{Value} &= 12.3788 \\
\end{align*}
\]

\[
\begin{align*}
F_1 &= -0.6299*F_3 + 1.0000 \ D_1 \\
\text{Std Err} &= 0.0563 \ \text{Gam1} \\
t \ \text{Value} &= -11.1809 \\
F_2 &= 0.5931*F_1 - 0.2409*F_3 + 1.0000 \ D_2 \\
\text{Std Err} &= 0.0468 \ \text{Beta} 0.0549 \ \text{Gam2} \\
t \ \text{Value} &= 12.6788 -4.3885 \\
\end{align*}
\]
The measurement scale of variables is often arbitrary. Therefore, it can be useful to look at the standardized equations produced by PROC CALIS. Output 17.1.8 displays the standardized equations and predicted moments. From the standardized structural equations for \( F_1 \) and \( F_2 \), you can conclude that SES (\( F_3 \)) has a larger impact on earlier Alienation (\( F_1 \)) than on later Alienation (\( F_3 \)).

The squared multiple correlation for each equation, the correlation among the exogenous variables, and the covariance matrices among the latent variables and between the observed and the latent variables help to describe the relationships among all variables.
Output 17.1.8. Standardized Equations and Predicted Moments

Stability of Alienation
Data Matrix of WHEATON, MUTHEN, ALWIN & SUMMERS (1977)

The CALIS Procedure
Covariance Structure Analysis: Maximum Likelihood Estimation

\begin{align*}
V_1 &= 0.8348 F_1 + 0.5505 E_1 \\
V_2 &= 0.7846 F_1 + 0.6200 E_2 \\
V_3 &= 0.8450 F_2 + 0.5348 E_3 \\
V_4 &= 0.7968 F_2 + 0.6043 E_4 \\
V_5 &= 0.8297 F_3 + 0.5581 E_5 \\
V_6 &= 0.6508 F_3 + 0.7593 E_6
\end{align*}

\text{Lamb}

\begin{align*}
F_1 &= -0.5626 F_3 + 0.8268 D_1 \\
F_2 &= 0.5692 F_1 + -0.2064 F_3 + 0.7080 D_2
\end{align*}

\text{Gam1 Gam2}

Output 17.1.8. (continued)

Stability of Alienation
Data Matrix of WHEATON, MUTHEN, ALWIN & SUMMERS (1977)

The CALIS Procedure
Covariance Structure Analysis: Maximum Likelihood Estimation

Squared Multiple Correlations

<table>
<thead>
<tr>
<th>Variable</th>
<th>Error Variance</th>
<th>Total Variance</th>
<th>R-Square</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 V1</td>
<td>3.60788</td>
<td>11.90391</td>
<td>0.6969</td>
</tr>
<tr>
<td>2 V2</td>
<td>3.59493</td>
<td>9.35145</td>
<td>0.6156</td>
</tr>
<tr>
<td>3 V3</td>
<td>3.60788</td>
<td>12.61575</td>
<td>0.7140</td>
</tr>
<tr>
<td>4 V4</td>
<td>3.59493</td>
<td>9.84539</td>
<td>0.6349</td>
</tr>
<tr>
<td>5 V5</td>
<td>2.99368</td>
<td>9.61000</td>
<td>0.6885</td>
</tr>
<tr>
<td>6 V6</td>
<td>259.57580</td>
<td>450.28800</td>
<td>0.4235</td>
</tr>
<tr>
<td>7 F1</td>
<td>5.67047</td>
<td>8.29603</td>
<td>0.3165</td>
</tr>
<tr>
<td>8 F2</td>
<td>4.51480</td>
<td>9.00787</td>
<td>0.4988</td>
</tr>
</tbody>
</table>

Correlations Among Exogenous Variables

<table>
<thead>
<tr>
<th>Var1 Var2 Parameter</th>
<th>Estimate</th>
</tr>
</thead>
<tbody>
<tr>
<td>E1 E3 The5</td>
<td>0.25106</td>
</tr>
<tr>
<td>E2 E4 The5</td>
<td>0.25197</td>
</tr>
</tbody>
</table>
Output 17.1.8. (continued)

<table>
<thead>
<tr>
<th></th>
<th>F1</th>
<th>F2</th>
<th>F3</th>
</tr>
</thead>
<tbody>
<tr>
<td>F1</td>
<td>8.296026985</td>
<td>5.924364730</td>
<td>-4.167911571</td>
</tr>
<tr>
<td>F2</td>
<td>5.924364730</td>
<td>9.007870649</td>
<td>-4.065656060</td>
</tr>
<tr>
<td>F3</td>
<td>-4.167911571</td>
<td>-4.065656060</td>
<td>6.616317547</td>
</tr>
</tbody>
</table>

Output 17.1.9 displays the latent variable score regression coefficients that produce the latent variable scores. Each latent variable is expressed as a linear combination of the observed variables. See Chapter 52, “The SCORE Procedure,” for more information on the creation of latent variable scores. Note that the total effects and indirect effects of the exogenous variables are also displayed.
### Example 17.1. Path Analysis: Stability of Alienation

**Output 17.1.9.** Latent Variable Score Regression, Direct and Indirect Effects

#### Stability of Alienation
Data Matrix of WHEATON, MUTHEN, ALWIN & SUMMERS (1977)

The CALIS Procedure
Covariance Structure Analysis: Maximum Likelihood Estimation

Latent Variable Score Regression Coefficients

<table>
<thead>
<tr>
<th></th>
<th>F1</th>
<th>F2</th>
<th>F3</th>
</tr>
</thead>
<tbody>
<tr>
<td>V1 Anomia (1967)</td>
<td>0.413</td>
<td>0.048</td>
<td>-0.052</td>
</tr>
<tr>
<td>V2 Anomia (1971)</td>
<td>0.345</td>
<td>0.040</td>
<td>-0.043</td>
</tr>
<tr>
<td>V3 Education</td>
<td>0.053</td>
<td>0.431</td>
<td>-0.039</td>
</tr>
<tr>
<td>V4 Powerlessness (1967)</td>
<td>0.044</td>
<td>0.360</td>
<td>-0.034</td>
</tr>
<tr>
<td>V5 Powerlessness (1971)</td>
<td>-0.075</td>
<td>-0.064</td>
<td>0.506</td>
</tr>
<tr>
<td>V6 Occupational Status Index</td>
<td>-0.005</td>
<td>0.000</td>
<td>0.000</td>
</tr>
</tbody>
</table>

Total Effects of Exogenous on Endogenous Variables

<table>
<thead>
<tr>
<th></th>
<th>F3</th>
<th>F1</th>
<th>F2</th>
</tr>
</thead>
<tbody>
<tr>
<td>V1</td>
<td>-0.630</td>
<td>1.000</td>
<td>0.000</td>
</tr>
<tr>
<td>V2</td>
<td>-0.525</td>
<td>0.833</td>
<td>0.000</td>
</tr>
<tr>
<td>V3</td>
<td>-0.614</td>
<td>0.593</td>
<td>1.000</td>
</tr>
<tr>
<td>V4</td>
<td>-0.512</td>
<td>0.494</td>
<td>0.833</td>
</tr>
<tr>
<td>V5</td>
<td>1.000</td>
<td>0.000</td>
<td>0.000</td>
</tr>
<tr>
<td>V6</td>
<td>5.369</td>
<td>0.000</td>
<td>0.000</td>
</tr>
<tr>
<td>F1</td>
<td>-0.629</td>
<td>0.000</td>
<td>0.000</td>
</tr>
<tr>
<td>F2</td>
<td>-0.614</td>
<td>0.593</td>
<td>0.000</td>
</tr>
</tbody>
</table>

Indirect Effects of Exogenous on Endogenous Variables

<table>
<thead>
<tr>
<th></th>
<th>F3</th>
<th>F1</th>
<th>F2</th>
</tr>
</thead>
<tbody>
<tr>
<td>V1</td>
<td>-0.629</td>
<td>0.000</td>
<td>0.000</td>
</tr>
<tr>
<td>V2</td>
<td>-0.524</td>
<td>0.000</td>
<td>0.000</td>
</tr>
<tr>
<td>V3</td>
<td>-0.614</td>
<td>0.000</td>
<td>0.000</td>
</tr>
<tr>
<td>V4</td>
<td>-0.512</td>
<td>0.000</td>
<td>0.000</td>
</tr>
<tr>
<td>V5</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
</tr>
<tr>
<td>V6</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
</tr>
<tr>
<td>F1</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
</tr>
<tr>
<td>F2</td>
<td>-0.373</td>
<td>0.000</td>
<td>0.000</td>
</tr>
</tbody>
</table>

PROC CALIS can display Lagrange multiplier and Wald statistics for model modifications. Modification indices are displayed for each parameter matrix. Only the Lagrange multiplier statistics have significance levels and approximate changes of values displayed. The significance level of the Wald statistic for a given parameter is the same as that shown in the equation output. An insignificant $p$-value for a Wald statistic means that the corresponding parameter can be dropped from the model without significantly worsening the fit of the model.

A significant $p$-value for a Lagrange multiplier test indicates that the model would achieve a better fit if the corresponding parameter is free. To aid in determining significant results, PROC CALIS displays the rank order of the ten largest Lagrange multiplier statistics. For example, [E5:E2] in the _PHI_ matrix is associated with the largest Lagrange multiplier statistic; the associated $p$-value is 0.0067. This means that adding a parameter for the covariance between E5 and E2 will lead to a significantly
better fit of the model. However, adding parameters indiscriminately can result in specification errors. An over-fitted model may not perform well with future samples. As always, the decision to add parameters should be accompanied with consideration and knowledge of the application area.

Output 17.1.10. Lagrange Multiplier and Wald Tests

Stability of Alienation
Data Matrix of WHEATON, MUTHEN, ALWIN & SUMMERS (1977)

The CALIS Procedure
Covariance Structure Analysis: Maximum Likelihood Estimation

Lagrange Multiplier and Wald Test Indices \( \Phi_{[9:9]} \)
Symmetric Matrix

Univariate Tests for Constant Constraints
Lagrange Multiplier or Wald Index / Probability / Approx Change of Value

<p>| | | | | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>F3</td>
<td>E1</td>
<td>E2</td>
<td>E3</td>
<td>E4</td>
<td>E5</td>
<td>E6</td>
<td>D1</td>
<td>D2</td>
</tr>
<tr>
<td>-------</td>
<td>-------</td>
<td>-------</td>
<td>-------</td>
<td>-------</td>
<td>-------</td>
<td>-------</td>
<td>-------</td>
<td>-------</td>
</tr>
<tr>
<td>F3</td>
<td>107.1619</td>
<td>3.3903</td>
<td>3.3901</td>
<td>0.5752</td>
<td>0.5753</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>0.0656</td>
<td>0.0656</td>
<td>0.4482</td>
<td>0.4482</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>[Phi]</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>0.5079</td>
<td>-0.4231</td>
<td>0.2090</td>
<td>-0.1741</td>
</tr>
<tr>
<td>E1</td>
<td>3.3903</td>
<td>322.4501</td>
<td>0.1529</td>
<td>55.4237</td>
<td>1.2037</td>
<td>5.8025</td>
<td>0.7398</td>
<td>0.4840</td>
</tr>
<tr>
<td></td>
<td>0.0656</td>
<td></td>
<td>0.6958</td>
<td></td>
<td>0.2726</td>
<td>0.0160</td>
<td>0.3897</td>
<td>0.4866</td>
</tr>
<tr>
<td></td>
<td>0.5079</td>
<td>0.0900</td>
<td></td>
<td></td>
<td>-0.3262</td>
<td>0.5193</td>
<td>-1.2587</td>
<td>0.2276</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>0.0001</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>E2</td>
<td>3.3901</td>
<td>0.1529</td>
<td>477.6768</td>
<td>0.5946</td>
<td>55.4237</td>
<td>7.3649</td>
<td>1.4168</td>
<td>0.4840</td>
</tr>
<tr>
<td></td>
<td>0.0656</td>
<td>0.6958</td>
<td></td>
<td>0.4406</td>
<td></td>
<td>0.0067</td>
<td>0.2339</td>
<td>0.4866</td>
</tr>
<tr>
<td></td>
<td>0.4482</td>
<td>0.0900</td>
<td></td>
<td>0.2328</td>
<td></td>
<td>-0.5060</td>
<td>1.5431</td>
<td>-1.896</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>E3</td>
<td>0.5752</td>
<td>55.4237</td>
<td>0.5946</td>
<td>322.4501</td>
<td>0.1528</td>
<td>1.5982</td>
<td>0.0991</td>
<td>1.1825</td>
</tr>
<tr>
<td></td>
<td>0.4482</td>
<td></td>
<td></td>
<td>0.6958</td>
<td>0.2062</td>
<td>0.7529</td>
<td>0.2768</td>
<td>0.4408</td>
</tr>
<tr>
<td></td>
<td>0.2090</td>
<td></td>
<td></td>
<td>0.2328</td>
<td></td>
<td>0.6958</td>
<td>0.2062</td>
<td>0.4408</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>1.896</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>E4</td>
<td>0.5753</td>
<td>1.2037</td>
<td>55.4237</td>
<td>0.1528</td>
<td>477.6768</td>
<td>1.2044</td>
<td>0.0029</td>
<td>1.1825</td>
</tr>
<tr>
<td></td>
<td>0.2090</td>
<td></td>
<td></td>
<td>0.2328</td>
<td></td>
<td>0.2724</td>
<td>0.9568</td>
<td>0.2768</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>E5</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>E6</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>D1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>D2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Output 17.1.10. (continued)

Stability of Alienation
Data Matrix of WHEATON, MUTHEN, ALWIN & SUMMERS (1977)

The CALIS Procedure
Covariance Structure Analysis: Maximum Likelihood Estimation

Rank Order of the 10 Largest Lagrange Multipliers in _PHI_

<table>
<thead>
<tr>
<th>Row</th>
<th>Column</th>
<th>Lagrange Multiplier</th>
<th>Pr &gt; Lagrange</th>
</tr>
</thead>
<tbody>
<tr>
<td>E5</td>
<td>E2</td>
<td>7.36486</td>
<td>0.0067</td>
</tr>
<tr>
<td>E5</td>
<td>E1</td>
<td>5.80246</td>
<td>0.0160</td>
</tr>
<tr>
<td>E1</td>
<td>F3</td>
<td>3.39030</td>
<td>0.0656</td>
</tr>
<tr>
<td>E2</td>
<td>F3</td>
<td>3.39013</td>
<td>0.0656</td>
</tr>
<tr>
<td>E5</td>
<td>E3</td>
<td>1.59820</td>
<td>0.2062</td>
</tr>
<tr>
<td>E6</td>
<td>E2</td>
<td>1.41677</td>
<td>0.2339</td>
</tr>
<tr>
<td>E5</td>
<td>E4</td>
<td>1.20437</td>
<td>0.2724</td>
</tr>
<tr>
<td>E4</td>
<td>E1</td>
<td>1.20367</td>
<td>0.2726</td>
</tr>
<tr>
<td>D1</td>
<td>E3</td>
<td>1.18251</td>
<td>0.2768</td>
</tr>
<tr>
<td>D1</td>
<td>E4</td>
<td>1.18249</td>
<td>0.2768</td>
</tr>
</tbody>
</table>
Output 17.1.10. (continued)

Stability of Alienation
Data Matrix of WHEATON, MUTHEN, ALWIN & SUMMERS (1977)

The CALIS Procedure
Covariance Structure Analysis: Maximum Likelihood Estimation
Lagrange Multiplier and Wald Test Indices _GAMMA_[8:1]
General Matrix
Univariate Tests for Constant Constraints
Lagrange Multiplier or Wald Index / Probability / Approx Change of Value

<table>
<thead>
<tr>
<th></th>
<th>Lagrange Multiplier</th>
<th>Wald Index</th>
<th>Probability</th>
<th>Approx Change of Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>F3</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>V1</td>
<td>3.3903</td>
<td>0.0656</td>
<td>0.0768</td>
<td></td>
</tr>
<tr>
<td>V2</td>
<td>3.3901</td>
<td>0.0656</td>
<td>-0.0639</td>
<td></td>
</tr>
<tr>
<td>V3</td>
<td>0.5752</td>
<td>0.4482</td>
<td>0.0316</td>
<td></td>
</tr>
<tr>
<td>V4</td>
<td>0.5753</td>
<td>0.4482</td>
<td>-0.0263</td>
<td></td>
</tr>
<tr>
<td>V5</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>Sing</td>
</tr>
<tr>
<td>V6</td>
<td>153.2354</td>
<td>.</td>
<td>.</td>
<td>[Lamb]</td>
</tr>
<tr>
<td>F1</td>
<td>125.0132</td>
<td>.</td>
<td>.</td>
<td>[Gam1]</td>
</tr>
<tr>
<td>F2</td>
<td>19.2585</td>
<td>.</td>
<td>.</td>
<td>[Gam2]</td>
</tr>
</tbody>
</table>
### Output 17.1.10. (continued)

Stability of Alienation  
Data Matrix of WHEATON, MUTHEN, ALWIN & SUMMERS (1977)  

The CALIS Procedure  
Covariance Structure Analysis: Maximum Likelihood Estimation  

**Rank Order of the 4 Largest Lagrange Multipliers in \( \gamma \)**

<table>
<thead>
<tr>
<th>Row</th>
<th>Column</th>
<th>Lagrange Multiplier</th>
<th>Pr &gt; Lagrange</th>
</tr>
</thead>
<tbody>
<tr>
<td>V1</td>
<td>F3</td>
<td>3.39030</td>
<td>0.0656</td>
</tr>
<tr>
<td>V2</td>
<td>F3</td>
<td>3.39013</td>
<td>0.0656</td>
</tr>
<tr>
<td>V4</td>
<td>F3</td>
<td>0.57526</td>
<td>0.4482</td>
</tr>
<tr>
<td>V3</td>
<td>F3</td>
<td>0.57523</td>
<td>0.4482</td>
</tr>
</tbody>
</table>
Output 17.1.10.  (continued)

### Stability of Alienation

Data Matrix of WHEATON, MUTHEN, ALWIN & SUMMERS (1977)

The CALIS Procedure

Covariance Structure Analysis: Maximum Likelihood Estimation

Lagrange Multiplier and Wald Test Indices _BETA_[8:8]

General Matrix

Identity-Minus-Inverse Model Matrix

Univariate Tests for Constant Constraints

<table>
<thead>
<tr>
<th>Lagrange Multiplier or Wald Index</th>
<th>Probability</th>
<th>Approx Change of Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>V1</td>
<td>V2</td>
<td>V3</td>
</tr>
<tr>
<td>0.5957</td>
<td>.</td>
<td>0.6406</td>
</tr>
<tr>
<td>0.4402</td>
<td>.</td>
<td>0.4235</td>
</tr>
<tr>
<td>0.0218</td>
<td>.</td>
<td>0.0185</td>
</tr>
</tbody>
</table>

### Sing

| V2                               | V3          | V4                      | V5 | V6 | F1  | F2  |
| 0.5957                           | 0.3027      | 0.1446                  | 1.1537 | 0.0296 | 0.1588 | 0.0817 |
| 0.5355                           | 0.5822      | 0.7038                  | 0.2828 | 0.8634 | 0.6902 | 0.7750 |
| 0.0178                           | 0.0180      | -0.0145                 | 0.0322 | 0.0007 | 0.0144 | -0.0110 |

### Sing

| V3                               | V4          | V5                      | V6 | F1  | F2  |
| 0.4487                           | 0.2519      | 0.0002                  | 0.9867 | 0.1442 | 0.1588 | 0.0817 |
| 0.5030                           | 0.6157      | 0.9877                  | 0.3206 | 0.7041 | 0.6903 | 0.7750 |
| -0.0160                          | -0.0144     | -0.0004                 | -0.0249 | -0.0014 | -0.0120 | 0.0092 |

### Sing

| V4                               | V5          | V6                      | F1  | F2  |
| 0.4085                           | 8.6455      | 2.7123                  | 1.6147 | 1.6128 | 0.1032 | 0.1035 |
| 0.0200                           | 0.0033      | 0.0996                  | 0.1430 | 0.7479 | 0.7476 |
| 0.1242                           | -0.1454     | 0.0785                  | -0.0674 | .     | .     | -0.0490 | 0.0329 |

### Sing

| V5                               | V6          | F1  | F2  |
| 0.4209                           | 1.4387      | 1.034     | 0.1035     |
| 0.5165                           | 0.2304      | 0.4878     | 0.7477     |
| -0.2189                          | 0.3924      | 0.2629     | -0.1765    |

### Sing

| F1                               | F2          |
| 1.0998                           | 1.1021      |
| 0.2943                           | 0.2938      |
| 0.0977                           | -0.0817     |

### Sing

| F2                               |
| 0.0193                           |
| 0.8896                           |
| -0.0104                          |
When you specify equality constraints, PROC CALIS displays Lagrange multiplier tests for releasing the constraints. In the current example, none of the three constraints achieve a $p$-value smaller than 0.05. This means that releasing the constraints may not lead to a significantly better fit of the model. Therefore, all constraints are retained in the model.

Output 17.1.11. Tests for Equality Constraints

The model is specified using the LINEQS, STD, and COV statements. The section “Getting Started” on page 407 also contains the COSAN and RAM specifications of this model. These model specifications would give essentially the same results.
Example 17.2. Simultaneous Equations with Intercept

The demand-and-supply food example of Kmenta (1971, pp. 565, 582) is used to illustrate the use of PROC CALIS for the estimation of intercepts and coefficients of simultaneous equations. The model is specified by two simultaneous equations containing two endogenous variables $Q$ and $P$ and three exogenous variables $D$, $F$, and $Y$.

$$Q_{t\text{ (demand)}} = \alpha_1 + \beta_1 P_t + \gamma_1 D_t$$

$$Q_{t\text{ (supply)}} = \alpha_2 + \beta_2 P_t + \gamma_2 F_t + \gamma_3 Y_t$$

for $t = 1, \ldots, 20$. 
The LINEQS statement requires that each endogenous variable appear on the left-hand side of exactly one equation. Instead of analyzing the system

$$\mathbf{B}^* \eta = \Gamma \xi + \zeta$$

PROC CALIS analyzes the equivalent system

$$\eta = \mathbf{B} \eta + \Gamma \xi + \zeta$$

with $\mathbf{B}^* = \mathbf{I} - \mathbf{B}$. This requires that one of the preceding equations be solved for $P_t$. Solving the second equation for $P_t$ yields

$$P_t = \frac{1}{\beta_2} Q_t - \frac{\alpha_2}{\beta_2} F_t - \frac{\gamma_2}{\beta_2} Y_t - \frac{\gamma_3}{\beta_2} Y_t$$

You can estimate the intercepts of a system of simultaneous equations by applying PROC CALIS on the uncorrected covariance (UCOV) matrix of the data set that is augmented by an additional constant variable with the value 1. In the following example, the uncorrected covariance matrix is augmented by an additional variable INTERCEPT by using the AUGMENT option. The PROC CALIS statement contains the options UCOV and AUG to compute and analyze an augmented UCOV matrix from the input data set FOOD.

```sas
Data food;
Title 'Food example of KMENTA(1971, p.565 & 582)';
Input Q P D F Y;
Label Q='Food Consumption per Head'
   P='Ratio of Food Prices to General Price'
   D='Disposable Income in Constant Prices'
   F='Ratio of Preceding Years Prices'
   Y='Time in Years 1922-1941';
datalines;
98.485 100.323 87.4 98.0 1
99.187 104.264 97.6 99.1 2
102.163 103.435 96.7 99.1 3
101.504 104.506 98.2 98.1 4
104.240 98.001 99.8 110.8 5
103.243 99.456 100.5 108.2 6
103.993 101.066 103.2 105.6 7
99.900 104.763 107.8 109.8 8
100.350 96.446 96.6 108.7 9
95.435 93.085 75.1 81.0 11
92.424 98.801 76.9 68.6 12
94.535 102.908 84.6 70.9 13
98.757 98.756 90.6 81.4 14
105.797 95.119 103.1 102.3 15
100.225 98.451 105.1 105.0 16
103.522 86.498 96.4 110.5 17
99.929 104.016 104.4 92.5 18
```
Chapter 17. The CALIS Procedure

The following, essentially equivalent model definition uses program code to reparameterize the model in terms of the original equations; the output is displayed in Output 17.2.1.

```sas
proc calis data=food ucov aug pshort;
   Lineqs
      Q = alf1 Intercept + alf2 P + alf3 D + E1,
      P = gam1 Intercept + gam2 Q + gam3 F + gam4 Y + E2;
   Std
      E1-E2 = eps1-eps2;
   Cov
      E1-E2 = eps3;
   Bounds
      eps1-eps2 >= 0. ;
run;
```

The following, essentially equivalent model definition uses program code to reparameterize the model in terms of the original equations; the output is displayed in Output 17.2.1.

```sas
proc calis data=food ucov aug pshort;
   Lineqs
      Q = alphal Intercept + beta1 P + gamma1 D + E1,
      P = alpha2_b Intercept + gamma2_b F + gamma3_b Y + _b Q + E2;
   Std
      E1-E2 = eps1-eps2;
   Cov
      E1-E2 = eps3;
   Parameters alpha2 (50.) beta2 gamma2 gamma3 (3*.25);
      alpha2_b = -alpha2 / beta2;
      gamma2_b = -gamma2 / beta2;
      gamma3_b = -gamma3 / beta2;
      _b = 1 / beta2;
   Bounds
      eps1-eps2 >= 0. ;
run;
```
Output 17.2.1.  Food Example of Kmenta

<table>
<thead>
<tr>
<th>Term</th>
<th>Matrix</th>
<th>Rows</th>
<th>Columns</th>
<th>Matrix Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td><em>SEL</em></td>
<td>6</td>
<td>8</td>
<td>SELECTION</td>
</tr>
<tr>
<td>2</td>
<td><em>BETA</em></td>
<td>8</td>
<td>8</td>
<td>EQSBETA</td>
</tr>
<tr>
<td>3</td>
<td><em>GAMMA</em></td>
<td>8</td>
<td>6</td>
<td>EQSGAMMA</td>
</tr>
<tr>
<td>4</td>
<td><em>PHI</em></td>
<td>6</td>
<td>6</td>
<td>SYMMETRIC</td>
</tr>
</tbody>
</table>

The 2 Endogenous Variables

<table>
<thead>
<tr>
<th>Manifest</th>
<th>Q</th>
<th>P</th>
</tr>
</thead>
<tbody>
<tr>
<td>Latent</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

The 6 Exogenous Variables

<table>
<thead>
<tr>
<th>Manifest</th>
<th>D</th>
<th>F</th>
<th>Y</th>
<th>Intercept</th>
</tr>
</thead>
<tbody>
<tr>
<td>Latent</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Error</td>
<td>E1</td>
<td>E2</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Food example of KMENTA (1971, p. 565 & 582)

The CALIS Procedure
Covariance Structure Analysis: Maximum Likelihood Estimation

Parameter Estimates 10
Functions (Observations) 21
Lower Bounds 2
Upper Bounds 0

Optimization Start
Active Constraints 0
Max Abs Gradient Element 203.9741437
Radius 62167.829174

Ratio Between Actual and Predicted Objectives

<table>
<thead>
<tr>
<th>Iter</th>
<th>Restarts</th>
<th>Calls</th>
<th>Active Constraints</th>
<th>Objective Function</th>
<th>Gradient Element</th>
<th>Lambda</th>
<th>Ratio</th>
<th>Change</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0</td>
<td>2</td>
<td>0</td>
<td>1.19094</td>
<td>1.1591</td>
<td>3.9410</td>
<td>0</td>
<td>0.688</td>
</tr>
<tr>
<td>2</td>
<td>0</td>
<td>5</td>
<td>0</td>
<td>0.32678</td>
<td>0.8642</td>
<td>9.9864</td>
<td>0.00127</td>
<td>2.356</td>
</tr>
<tr>
<td>3</td>
<td>0</td>
<td>7</td>
<td>0</td>
<td>0.19108</td>
<td>0.1357</td>
<td>5.5100</td>
<td>0.00006</td>
<td>0.685</td>
</tr>
<tr>
<td>4</td>
<td>0</td>
<td>10</td>
<td>0</td>
<td>0.16682</td>
<td>0.0243</td>
<td>2.0513</td>
<td>0.00005</td>
<td>0.867</td>
</tr>
<tr>
<td>5</td>
<td>0</td>
<td>12</td>
<td>0</td>
<td>0.16288</td>
<td>0.00393</td>
<td>1.0570</td>
<td>0.00014</td>
<td>0.828</td>
</tr>
<tr>
<td>6</td>
<td>0</td>
<td>13</td>
<td>0</td>
<td>0.16132</td>
<td>0.00156</td>
<td>0.3643</td>
<td>0.00004</td>
<td>0.864</td>
</tr>
<tr>
<td>7</td>
<td>0</td>
<td>15</td>
<td>0</td>
<td>0.16077</td>
<td>0.000557</td>
<td>0.2176</td>
<td>0.00006</td>
<td>0.984</td>
</tr>
<tr>
<td>8</td>
<td>0</td>
<td>16</td>
<td>0</td>
<td>0.16052</td>
<td>0.000250</td>
<td>0.1819</td>
<td>0.00001</td>
<td>0.618</td>
</tr>
<tr>
<td>9</td>
<td>0</td>
<td>17</td>
<td>0</td>
<td>0.16032</td>
<td>0.000201</td>
<td>0.0662</td>
<td>0</td>
<td>0.971</td>
</tr>
<tr>
<td>10</td>
<td>0</td>
<td>18</td>
<td>0</td>
<td>0.16030</td>
<td>0.000011</td>
<td>0.0195</td>
<td>0</td>
<td>1.108</td>
</tr>
<tr>
<td>11</td>
<td>0</td>
<td>19</td>
<td>0</td>
<td>0.16030</td>
<td>6.116E-7</td>
<td>0.00763</td>
<td>0</td>
<td>1.389</td>
</tr>
<tr>
<td>12</td>
<td>0</td>
<td>20</td>
<td>0</td>
<td>0.16030</td>
<td>9.454E-8</td>
<td>0.00301</td>
<td>0</td>
<td>1.389</td>
</tr>
<tr>
<td>13</td>
<td>0</td>
<td>21</td>
<td>0</td>
<td>0.16030</td>
<td>1.461E-8</td>
<td>0.00118</td>
<td>0</td>
<td>1.388</td>
</tr>
<tr>
<td>14</td>
<td>0</td>
<td>22</td>
<td>0</td>
<td>0.16030</td>
<td>2.269E-9</td>
<td>0.000465</td>
<td>0</td>
<td>1.395</td>
</tr>
<tr>
<td>15</td>
<td>0</td>
<td>23</td>
<td>0</td>
<td>0.16030</td>
<td>3.59E-10</td>
<td>0.000182</td>
<td>0</td>
<td>1.427</td>
</tr>
</tbody>
</table>

Optimization Results

<table>
<thead>
<tr>
<th>Iterations</th>
<th>Function Calls</th>
<th>Jacobian Calls</th>
<th>Objective Function</th>
<th>Max Abs Gradient Element</th>
<th>Lambda</th>
<th>Actual Over Pred Change</th>
</tr>
</thead>
<tbody>
<tr>
<td>15</td>
<td>24</td>
<td>16</td>
<td>0.1603035477</td>
<td>0.001820805</td>
<td>0.00010322573</td>
<td>1.4266532872</td>
</tr>
</tbody>
</table>

GCONV convergence criterion satisfied.
### Output 17.2.1. (continued)

#### Food example of KMENTA (1971, p. 565 & 582)

<table>
<thead>
<tr>
<th>Method</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fit Function</td>
<td>0.1603</td>
</tr>
<tr>
<td>Goodness of Fit Index (GFI)</td>
<td>0.9530</td>
</tr>
<tr>
<td>GFI Adjusted for Degrees of Freedom (AGFI)</td>
<td>0.0120</td>
</tr>
<tr>
<td>Root Mean Square Residual (RMR)</td>
<td>2.0653</td>
</tr>
<tr>
<td>Parsimonious GFI (Mulaik, 1989)</td>
<td>0.0635</td>
</tr>
<tr>
<td>Chi-Square</td>
<td>3.0458</td>
</tr>
<tr>
<td>Chi-Square DF</td>
<td>1</td>
</tr>
<tr>
<td>Pr &gt; Chi-Square</td>
<td>0.0809</td>
</tr>
<tr>
<td>Independence Model Chi-Square</td>
<td>534.27</td>
</tr>
<tr>
<td>Independence Model Chi-Square DF</td>
<td>15</td>
</tr>
<tr>
<td>RMSEA Estimate</td>
<td>0.3281</td>
</tr>
<tr>
<td>RMSEA 90% Lower Confidence Limit</td>
<td>0.7777</td>
</tr>
<tr>
<td>RMSEA 90% Upper Confidence Limit</td>
<td>1.8270</td>
</tr>
<tr>
<td>ECVI Estimate</td>
<td>3.3493</td>
</tr>
<tr>
<td>Probability of Close Fit</td>
<td>0.0882</td>
</tr>
<tr>
<td>Bentler’s Comparative Fit Index</td>
<td>0.9961</td>
</tr>
<tr>
<td>Normal Theory Reweighted LS Chi-Square</td>
<td>2.8142</td>
</tr>
<tr>
<td>Akaike’s Information Criterion</td>
<td>1.0458</td>
</tr>
<tr>
<td>Bozdogan’s (1987) CAIC</td>
<td>-0.9500</td>
</tr>
<tr>
<td>Schwarz’s Bayesian Criterion</td>
<td>0.0500</td>
</tr>
<tr>
<td>McDonald’s (1989) Centrality</td>
<td>0.9501</td>
</tr>
<tr>
<td>Bentler &amp; Bonett’s (1980) Non-normed Index</td>
<td>0.9409</td>
</tr>
<tr>
<td>Bentler &amp; Bonett’s (1980) NFI</td>
<td>0.9943</td>
</tr>
<tr>
<td>James, Mulaik, &amp; Brett (1982) Parsimonious NFI</td>
<td>0.0663</td>
</tr>
<tr>
<td>Z-Test of Wilson &amp; Hilferty (1931)</td>
<td>1.4250</td>
</tr>
<tr>
<td>Bollen (1986) Normed Index Rh1</td>
<td>0.9145</td>
</tr>
<tr>
<td>Bollen (1988) Non-normed Index Delta2</td>
<td>0.9962</td>
</tr>
<tr>
<td>Hoelter’s (1983) Critical N</td>
<td>25</td>
</tr>
</tbody>
</table>

#### The CALIS Procedure

Covariance Structure Analysis: Maximum Likelihood Estimation

<table>
<thead>
<tr>
<th>Equation</th>
<th>Coefficient</th>
</tr>
</thead>
<tbody>
<tr>
<td>Q = -0.2295<em>P + 0.3100</em>D + 93.6193*Intercept + 1.0000 E1</td>
<td>beta1 gamma1 alphas1</td>
</tr>
<tr>
<td>P = 4.2140<em>Q - 0.9305</em>F - 1.5579<em>Y + 218.9</em>Intercept + 1.0000 E2</td>
<td>gamma2_b gamma3_b alpha2_b</td>
</tr>
</tbody>
</table>
Output 17.2.1.  (continued)

Food example of KMENTA(1971, p.565 & 582)

The CALIS Procedure
Covariance Structure Analysis: Maximum Likelihood Estimation

Variance of Exogenous Variables

<table>
<thead>
<tr>
<th>Variable</th>
<th>Parameter</th>
<th>Estimate</th>
</tr>
</thead>
<tbody>
<tr>
<td>D</td>
<td></td>
<td>10154</td>
</tr>
<tr>
<td>F</td>
<td></td>
<td>9989</td>
</tr>
<tr>
<td>Y</td>
<td></td>
<td>151.05263</td>
</tr>
<tr>
<td>Intercept</td>
<td></td>
<td>1.05263</td>
</tr>
<tr>
<td>E1</td>
<td>eps1</td>
<td>3.51274</td>
</tr>
<tr>
<td>E2</td>
<td>eps2</td>
<td>105.06746</td>
</tr>
</tbody>
</table>

Covariances Among Exogenous Variables

<table>
<thead>
<tr>
<th>Var1</th>
<th>Var2</th>
<th>Parameter</th>
<th>Estimate</th>
</tr>
</thead>
<tbody>
<tr>
<td>D</td>
<td>F</td>
<td></td>
<td>9994</td>
</tr>
<tr>
<td>D</td>
<td>Y</td>
<td></td>
<td>1101</td>
</tr>
<tr>
<td>F</td>
<td>Y</td>
<td></td>
<td>1046</td>
</tr>
<tr>
<td>D</td>
<td>Intercept</td>
<td></td>
<td>102.66842</td>
</tr>
<tr>
<td>F</td>
<td>Intercept</td>
<td></td>
<td>101.71053</td>
</tr>
<tr>
<td>Y</td>
<td>Intercept</td>
<td></td>
<td>11.05263</td>
</tr>
<tr>
<td>E1</td>
<td>E2</td>
<td>eps3</td>
<td>-18.87270</td>
</tr>
</tbody>
</table>
You can obtain almost equivalent results by applying the SAS/ETS procedure SYSLIN on this problem.

Example 17.3. Second-Order Confirmatory Factor Analysis

A second-order confirmatory factor analysis model is applied to a correlation matrix of Thurstone reported by McDonald (1985). Using the LINEQS statement, the three-term second-order factor analysis model is specified in equations notation. The first-order loadings for the three factors, F1, F2, and F3, each refer to three variables, X1-X3, X4-X6, and X7-X9. One second-order factor, F4, reflects the correlations among the three first-order factors. The second-order factor correlation matrix P is defined as a 1 × 1 identity matrix. Choosing the second-order uniqueness matrix U2 as a diagonal matrix with parameters U21-U23 gives an unidentified model. To compute identified maximum likelihood estimates, the matrix U2 is defined as a 3 × 3
identity matrix. The following code generates results that are partially displayed in Output 17.3.1:

```sas
Data Thurst(TYPE=CORR);
Title "Example of THURSTONE resp. McDONALD (1985, p.57, p.105)"
   _TYPE_ = 'CORR'; Input _NAME_ $ Obs1-Obs9;
Label Obs1='Sentences' Obs2='Vocabulary' Obs3='Sentence Completion'
   Obs4='First Letters' Obs5='Four-letter Words' Obs6='Suffices'
   Obs7='Letter series' Obs8='Pedigrees' Obs9='Letter Grouping';
   datalines;
   Obs1 1. . . . .....
   Obs2 .828 1. . . . ....
   Obs3 .776 .779 1. . . . ....
   Obs4 .439 .493 .460 1. . . . ....
   Obs5 .432 .464 .425 .674 1. . . . ....
   Obs6 .447 .489 .443 .590 .541 1 . ....
   Obs7 .447 .432 .401 .381 .402 .288 1. . . ...
   Obs8 .541 .537 .534 .350 .367 .320 .555 1. . ....
   Obs9 .380 .358 .359 .424 .446 .325 .598 .452 1. . ....
   ;
   proc calis data=Thurst method=max edf=212 pestim se;
   Title2 "Identified Second Order Confirmatory Factor Analysis";
   Title3 "C = F1 * F2 * P * F1' + F1 * U2 * F1' + U1, With P=U2=Ide";
   Lineqs
   Obs1 = X1 F1 + E1,
   Obs2 = X2 F1 + E2,
   Obs3 = X3 F1 + E3,
   Obs4 = X4 F2 + E4,
   Obs5 = X5 F2 + E5,
   Obs6 = X6 F2 + E6,
   Obs7 = X7 F3 + E7,
   Obs8 = X8 F3 + E8,
   Obs9 = X9 F3 + E9,
   F1 = X10 F4 + E10,
   F2 = X11 F4 + E11,
   F3 = X12 F4 + E12;
   Std
   F4 = 1. ,
   E1-E9 = U11-U19 ,
   E10-E12 = 3 * 1.;
   Bounds
   0. <= U11-U19;
   run;
```
Example of THURSTONE resp. McDONALD (1985, p.57, p.105)
Identified Second Order Confirmatory Factor Analysis
\[ C = F1 \times F2 \times P \times F2' \times F1' + F1 \times U2 \times F1' + U1, \text{ with } P=U2=I \]

The CALIS Procedure
Covariance Structure Analysis: Maximum Likelihood Estimation

Parameter Estimates 21
Functions (Observations) 45
Lower Bounds 9
Upper Bounds 0

Optimization Start
Active Constraints 0 Objective Function 0.7151823452
Max Abs Gradient Element 0.406719803 Radius 2.25762496

<table>
<thead>
<tr>
<th>Iter</th>
<th>Restarts</th>
<th>Function Calls</th>
<th>Active Constraints</th>
<th>Objective Function</th>
<th>Objective Function Change</th>
<th>Max Abs Gradient Element</th>
<th>Lambda Change</th>
<th>Predicted Change</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0</td>
<td>2</td>
<td>0</td>
<td>0.23113</td>
<td>0.4840</td>
<td>0.1299</td>
<td>0</td>
<td>1.363</td>
</tr>
<tr>
<td>2</td>
<td>0</td>
<td>3</td>
<td>0</td>
<td>0.18322</td>
<td>0.0479</td>
<td>0.0721</td>
<td>0</td>
<td>1.078</td>
</tr>
<tr>
<td>3</td>
<td>0</td>
<td>4</td>
<td>0</td>
<td>0.18051</td>
<td>0.00271</td>
<td>0.0200</td>
<td>0</td>
<td>1.006</td>
</tr>
<tr>
<td>4</td>
<td>0</td>
<td>5</td>
<td>0</td>
<td>0.18022</td>
<td>0.000289</td>
<td>0.00834</td>
<td>0</td>
<td>1.093</td>
</tr>
<tr>
<td>5</td>
<td>0</td>
<td>6</td>
<td>0</td>
<td>0.18018</td>
<td>0.000041</td>
<td>0.00251</td>
<td>0</td>
<td>1.201</td>
</tr>
<tr>
<td>6</td>
<td>0</td>
<td>7</td>
<td>0</td>
<td>0.18017</td>
<td>6.523E-6</td>
<td>0.000114</td>
<td>0</td>
<td>1.289</td>
</tr>
<tr>
<td>7</td>
<td>0</td>
<td>8</td>
<td>0</td>
<td>0.18017</td>
<td>1.085E-6</td>
<td>0.000388</td>
<td>0</td>
<td>1.347</td>
</tr>
<tr>
<td>8</td>
<td>0</td>
<td>9</td>
<td>0</td>
<td>0.18017</td>
<td>1.853E-7</td>
<td>0.000173</td>
<td>0</td>
<td>1.380</td>
</tr>
<tr>
<td>9</td>
<td>0</td>
<td>10</td>
<td>0</td>
<td>0.18017</td>
<td>3.208E-8</td>
<td>0.000063</td>
<td>0</td>
<td>1.399</td>
</tr>
<tr>
<td>10</td>
<td>0</td>
<td>11</td>
<td>0</td>
<td>0.18017</td>
<td>5.593E-9</td>
<td>0.000028</td>
<td>0</td>
<td>1.408</td>
</tr>
<tr>
<td>11</td>
<td>0</td>
<td>12</td>
<td>0</td>
<td>0.18017</td>
<td>9.79E-10</td>
<td>0.000011</td>
<td>0</td>
<td>1.414</td>
</tr>
</tbody>
</table>

Optimization Results
Iterations 11 Function Calls 13
Jacobian Calls 12 Active Constraints 0
Objective Function 0.1801712147 Max Abs Gradient Element 0.0000105805
Lambda 0 Actual Over Pred Change 1.413587595
Radius 0.0002026369

GCONV convergence criterion satisfied.
Output 17.3.1. (continued)

Example of THURSTONE resp. McDONALD (1985, p. 57, p. 105)
Identified Second Order Confirmatory Factor Analysis
\[ C = F_1 \times F_2 \times P \times F_2' \times F_1' + F_1 \times U_2 \times F_1' + U_1, \text{ with } P=U_2=Ide \]

The CALIS Procedure
Covariance Structure Analysis: Maximum Likelihood Estimation

<table>
<thead>
<tr>
<th>Statistic</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fit Function</td>
<td>0.1802</td>
</tr>
<tr>
<td>Goodness of Fit Index (GFI)</td>
<td>0.9596</td>
</tr>
<tr>
<td>GFI Adjusted for Degrees of Freedom (AGFI)</td>
<td>0.9242</td>
</tr>
<tr>
<td>Root Mean Square Residual (RMR)</td>
<td>0.0436</td>
</tr>
<tr>
<td>Parsimonious GFI (Mulaik, 1989)</td>
<td>0.6397</td>
</tr>
<tr>
<td>Chi-Square</td>
<td>38.1963</td>
</tr>
<tr>
<td>Chi-Square DF</td>
<td>24</td>
</tr>
<tr>
<td>Pr &gt; Chi-Square</td>
<td>0.0331</td>
</tr>
<tr>
<td>Independence Model Chi-Square</td>
<td>1101.9</td>
</tr>
<tr>
<td>Independence Model Chi-Square DF</td>
<td>36</td>
</tr>
<tr>
<td>RMSEA Estimate</td>
<td>0.0528</td>
</tr>
<tr>
<td>RMSEA 90% Lower Confidence Limit</td>
<td>0.0153</td>
</tr>
<tr>
<td>RMSEA 90% Upper Confidence Limit</td>
<td>0.0831</td>
</tr>
<tr>
<td>ECVI Estimate</td>
<td>0.3881</td>
</tr>
<tr>
<td>ECVI 90% Lower Confidence Limit</td>
<td></td>
</tr>
<tr>
<td>ECVI 90% Upper Confidence Limit</td>
<td>0.4888</td>
</tr>
<tr>
<td>Probability of Close Fit</td>
<td>0.4088</td>
</tr>
<tr>
<td>Bentler’s Comparative Fit Index</td>
<td>0.9867</td>
</tr>
<tr>
<td>Normal Theory Reweighted LS Chi-Square</td>
<td>40.1947</td>
</tr>
<tr>
<td>Akaike’s Information Criterion</td>
<td>-9.8037</td>
</tr>
<tr>
<td>Bozdogan’s (1987) CAIC</td>
<td>-114.4747</td>
</tr>
<tr>
<td>Schwarz’s Bayesian Criterion</td>
<td>-90.4747</td>
</tr>
<tr>
<td>McDonald’s (1989) Centrality</td>
<td>0.9672</td>
</tr>
<tr>
<td>Bentler &amp; Bonett’s (1980) Non-normed Index</td>
<td>0.9800</td>
</tr>
<tr>
<td>Bentler &amp; Bonett’s (1980) NFI</td>
<td>0.9653</td>
</tr>
<tr>
<td>James, Mulaik, &amp; Brett (1982) Parsimonious NFI</td>
<td>0.6436</td>
</tr>
<tr>
<td>Z-Test of Wilson &amp; Hilferty (1931)</td>
<td>1.8373</td>
</tr>
<tr>
<td>Bollen (1986) Normed Index Rhol</td>
<td>0.9480</td>
</tr>
<tr>
<td>Bollen (1988) Non-normed Index Delta2</td>
<td>0.9868</td>
</tr>
<tr>
<td>Hoelter’s (1983) Critical N</td>
<td>204</td>
</tr>
</tbody>
</table>
Output 17.3.1. (continued)

Example of THURSTONE resp. McDONALD (1985, p.57, p.105)
Identified Second Order Confirmatory Factor Analysis
\[ C = F_1 \times F_2 \times P \times F_2' \times F_1' + F_1 \times U_2 \times F_1' + U_1, \text{ With } P=U_2=Id_e \]

The CALIS Procedure
Covariance Structure Analysis: Maximum Likelihood Estimation

<table>
<thead>
<tr>
<th>Obs</th>
<th>Estimate</th>
<th>Std Err</th>
<th>t Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Obs1</td>
<td>0.5151*F1 + 1.0000 E1</td>
<td>0.0629 X1</td>
<td>8.1868</td>
</tr>
<tr>
<td>Obs2</td>
<td>0.5203*F1 + 1.0000 E2</td>
<td>0.0634 X2</td>
<td>8.2090</td>
</tr>
<tr>
<td>Obs3</td>
<td>0.4874*F1 + 1.0000 E3</td>
<td>0.0608 X3</td>
<td>8.0151</td>
</tr>
<tr>
<td>Obs4</td>
<td>0.5211*F2 + 1.0000 E4</td>
<td>0.0611 X4</td>
<td>8.5342</td>
</tr>
<tr>
<td>Obs5</td>
<td>0.4971*F2 + 1.0000 E5</td>
<td>0.0590 X5</td>
<td>8.4213</td>
</tr>
<tr>
<td>Obs6</td>
<td>0.4381*F2 + 1.0000 E6</td>
<td>0.0560 X6</td>
<td>7.8283</td>
</tr>
<tr>
<td>Obs7</td>
<td>0.4524*F3 + 1.0000 E7</td>
<td>0.0660 X7</td>
<td>6.8584</td>
</tr>
<tr>
<td>Obs8</td>
<td>0.4173*F3 + 1.0000 E8</td>
<td>0.0622 X8</td>
<td>6.7135</td>
</tr>
<tr>
<td>Obs9</td>
<td>0.4076*F3 + 1.0000 E9</td>
<td>0.0613 X9</td>
<td>6.6484</td>
</tr>
</tbody>
</table>

Example of THURSTONE resp. McDONALD (1985, p.57, p.105)
Identified Second Order Confirmatory Factor Analysis
\[ C = F_1 \times F_2 \times P \times F_2' \times F_1' + F_1 \times U_2 \times F_1' + U_1, \text{ With } P=U_2=Id_e \]

The CALIS Procedure
Covariance Structure Analysis: Maximum Likelihood Estimation

<table>
<thead>
<tr>
<th>F1</th>
<th>1.4438*F4 + 1.0000 E10</th>
<th>0.2565 X10</th>
<th>5.6282</th>
</tr>
</thead>
<tbody>
<tr>
<td>F2</td>
<td>1.2538*F4 + 1.0000 E11</td>
<td>0.2114 X11</td>
<td>5.9320</td>
</tr>
<tr>
<td>F3</td>
<td>1.4065*F4 + 1.0000 E12</td>
<td>0.2689 X12</td>
<td>5.2307</td>
</tr>
</tbody>
</table>
Example of THURSTONE resp. McDONALD (1985, p.57, p.105)
Identified Second Order Confirmatory Factor Analysis
C = F1 * F2 * P * F2’ * F1’ + F1 * U2 * F1’ + U1, With P=U2=Id

The CALIS Procedure
Covariance Structure Analysis: Maximum Likelihood Estimation

<table>
<thead>
<tr>
<th>Variable</th>
<th>Parameter</th>
<th>Estimate</th>
<th>Standard Error</th>
<th>t Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>F4</td>
<td></td>
<td>1.00000</td>
<td></td>
<td></td>
</tr>
<tr>
<td>E1</td>
<td>U11</td>
<td>0.18150</td>
<td>0.02848</td>
<td>6.37</td>
</tr>
<tr>
<td>E2</td>
<td>U12</td>
<td>0.16493</td>
<td>0.02777</td>
<td>5.94</td>
</tr>
<tr>
<td>E3</td>
<td>U13</td>
<td>0.26713</td>
<td>0.03336</td>
<td>8.01</td>
</tr>
<tr>
<td>E4</td>
<td>U14</td>
<td>0.30150</td>
<td>0.05102</td>
<td>5.91</td>
</tr>
<tr>
<td>E5</td>
<td>U15</td>
<td>0.36450</td>
<td>0.05264</td>
<td>6.93</td>
</tr>
<tr>
<td>E6</td>
<td>U16</td>
<td>0.50642</td>
<td>0.05963</td>
<td>8.49</td>
</tr>
<tr>
<td>E7</td>
<td>U17</td>
<td>0.39032</td>
<td>0.05934</td>
<td>6.58</td>
</tr>
<tr>
<td>E8</td>
<td>U18</td>
<td>0.48138</td>
<td>0.06225</td>
<td>7.73</td>
</tr>
<tr>
<td>E9</td>
<td>U19</td>
<td>0.50509</td>
<td>0.06333</td>
<td>7.98</td>
</tr>
<tr>
<td>E10</td>
<td></td>
<td>1.00000</td>
<td></td>
<td></td>
</tr>
<tr>
<td>E11</td>
<td></td>
<td>1.00000</td>
<td></td>
<td></td>
</tr>
<tr>
<td>E12</td>
<td></td>
<td>1.00000</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Example of THURSTONE resp. McDONALD (1985, p.57, p.105)
Identified Second Order Confirmatory Factor Analysis
\[ C = F_1 * F_2 * P * F_2' * F_1' + F_1 * U_2 * F_1' + U_1, \text{ With } P=U_2=I_{de} \]

The CALIS Procedure
Covariance Structure Analysis: Maximum Likelihood Estimation

\[
\begin{align*}
\text{Obs1} &= 0.9047*F_1 + 0.4260 E_1 \\
\text{Obs2} &= 0.9138*F_1 + 0.4061 E_2 \\
\text{Obs3} &= 0.8561*F_1 + 0.5168 E_3 \\
\text{Obs4} &= 0.8358*F_2 + 0.5491 E_4 \\
\text{Obs5} &= 0.7972*F_2 + 0.6037 E_5 \\
\text{Obs6} &= 0.7026*F_2 + 0.7116 E_6 \\
\text{Obs7} &= 0.7808*F_3 + 0.6248 E_7 \\
\text{Obs8} &= 0.7202*F_3 + 0.6938 E_8 \\
\text{Obs9} &= 0.7035*F_3 + 0.7107 E_9 \\
\end{align*}
\]

Example of THURSTONE resp. McDONALD (1985, p.57, p.105)
Identified Second Order Confirmatory Factor Analysis
\[ C = F_1 * F_2 * P * F_2' * F_1' + F_1 * U_2 * F_1' + U_1, \text{ With } P=U_2=I_{de} \]

The CALIS Procedure
Covariance Structure Analysis: Maximum Likelihood Estimation

\[
\begin{align*}
F_1 &= 0.8221*F_4 + 0.5694 E_{10} \\
F_2 &= 0.7818*F_4 + 0.6235 E_{11} \\
F_3 &= 0.8150*F_4 + 0.5794 E_{12} \\
\end{align*}
\]
Example of THURSTONE resp. MCDONALD (1985, p.57, p.105)
Identified Second Order Confirmatory Factor Analysis
C = F1 * F2 * P * F2' * F1' + F1 * U2 * F1' + U1, With P=U2=Ide

The CALIS Procedure
Covariance Structure Analysis: Maximum Likelihood Estimation

Squared Multiple Correlations

<table>
<thead>
<tr>
<th>Variable</th>
<th>Error Variance</th>
<th>Total Variance</th>
<th>R-Square</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 Obs1</td>
<td>0.18150</td>
<td>1.00000</td>
<td>0.8185</td>
</tr>
<tr>
<td>2 Obs2</td>
<td>0.16493</td>
<td>1.00000</td>
<td>0.8351</td>
</tr>
<tr>
<td>3 Obs3</td>
<td>0.26713</td>
<td>1.00000</td>
<td>0.7329</td>
</tr>
<tr>
<td>4 Obs4</td>
<td>0.30150</td>
<td>1.00000</td>
<td>0.6985</td>
</tr>
<tr>
<td>5 Obs5</td>
<td>0.36450</td>
<td>1.00000</td>
<td>0.6355</td>
</tr>
<tr>
<td>6 Obs6</td>
<td>0.50642</td>
<td>1.00000</td>
<td>0.4936</td>
</tr>
<tr>
<td>7 Obs7</td>
<td>0.39032</td>
<td>1.00000</td>
<td>0.6097</td>
</tr>
<tr>
<td>8 Obs8</td>
<td>0.48138</td>
<td>1.00000</td>
<td>0.5186</td>
</tr>
<tr>
<td>9 Obs9</td>
<td>0.50509</td>
<td>1.00000</td>
<td>0.4949</td>
</tr>
<tr>
<td>10 F1</td>
<td>1.00000</td>
<td>3.08452</td>
<td>0.6758</td>
</tr>
<tr>
<td>11 F2</td>
<td>1.00000</td>
<td>2.57213</td>
<td>0.6112</td>
</tr>
<tr>
<td>12 F3</td>
<td>1.00000</td>
<td>2.97832</td>
<td>0.6642</td>
</tr>
</tbody>
</table>

To compute McDonald’s unidentified model, you would have to change the STD and BOUNDS statements to include three more parameters:

<table>
<thead>
<tr>
<th>Std</th>
</tr>
</thead>
<tbody>
<tr>
<td>F4        = 1. ,</td>
</tr>
<tr>
<td>E1-E9     = U11-U19 ,</td>
</tr>
<tr>
<td>E10-E12   = U21-U23 ;</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Bounds</th>
</tr>
</thead>
<tbody>
<tr>
<td>0. &lt;= U11-U19,</td>
</tr>
<tr>
<td>0. &lt;= U21-U23;</td>
</tr>
</tbody>
</table>

The unidentified model is indicated in the output by an analysis of the linear dependencies in the approximate Hessian matrix (not shown). Because the information matrix is singular, standard errors are computed based on a Moore-Penrose inverse. The results computed by PROC CALIS differ from those reported by McDonald (1985). In the case of an unidentified model, the parameter estimates are not unique.

To specify the identified model using the COSAN model statement, you can use the following statements:

Title2 "Identified Second Order Confirmatory Factor Analysis Using COSAN";
Title3 "C = F1*F2*P*F2'*F1' + F1*U2*F1' + U1, With P=U2=Ide";
proc calis data=Thurst method=max edf=212 pestim se;
Cosan F1(3) * F2(1) * P(1,Ide) + F1(3) * U2(3,Ide) + U1(9,Dia);
Matrix F1
[ ,1] = X1-X3,
[ ,2] = 3 * 0. X4-X6,
[ ,3] = 6 * 0. X7-X9;
Example 17.4. Linear Relations Among Factor Loadings

The correlation matrix from Kinzer and Kinzer (N=326) is used by Guttman (1957) as an example that yields an approximate simplex. McDonald (1980) uses this data set as an example of factor analysis where he supposes that the loadings of the second factor are a linear function of the loadings on the first factor, for example

\[ b_{j2} = \alpha + \beta b_{j1}, \quad j = 1, \ldots, n \]

This example is also discussed in Browne (1982). The matrix specification of the model is

\[ C = F_{1}F_{1}' \]

with

\[ F_{1} = \begin{pmatrix} b_{11} & \alpha + \beta b_{11} & u_{11} \\ b_{21} & \alpha + \beta b_{21} & u_{22} \\ b_{31} & \alpha + \beta b_{31} & u_{33} \\ b_{41} & \alpha + \beta b_{41} & u_{44} \\ b_{51} & \alpha + \beta b_{51} & u_{55} \\ b_{61} & \alpha + \beta b_{61} & u_{66} \end{pmatrix} \]

This example is recomputed by PROC CALIS to illustrate a simple application of the COSAN model statement combined with program statements. This example also serves to illustrate the identification problem.

Data Kinzer(TYPE=CORR);
Title "Data Matrix of Kinzer & Kinzer, see GUTTMAN (1957)"
_TYPE_ = 'CORR'; INPUT _NAME_ $ Obs1-Obs6;
datalines;
Obs1 1.00 . . . . .
Obs2 .51 1.00 . . . .
Obs3 .46 .51 1.00 . . .
Obs4 .46 .47 .54 1.00 .
Obs5 .40 .39 .49 .57 1.00 .
Obs6 .33 .39 .47 .45 .56 1.00 ;
In a first test run of PROC CALIS, the same model is used as reported in McDonald (1980). Using the Levenberg-Marquardt optimization algorithm, this example specifies maximum likelihood estimation in the following code:

```sas
proc calis data=Kinzer method=max outram=ram nobs=326;
   title2 "Linearly Related Factor Analysis, (McDonald,1980)";
   title3 "Identification Problem";
   cosan F(8,Gen) * I(8,Ide);
   matrix F
      [ ,1]= X1-X6,
      [ ,2]= X7-X12,
      [1,3]= X13-X18;
   parms Alfa = .5 Beta = -.5;
   x7 = Alfa + Beta * X1;
   x8 = Alfa + Beta * X2;
   x9 = Alfa + Beta * X3;
   x10 = Alfa + Beta * X4;
   x11 = Alfa + Beta * X5;
   x12 = Alfa + Beta * X6;
   bounds X13-X18 >= 0.;
   vnames F Fact1 Fact2 Uvar1-Uvar6;
run;
```

The pattern of the initial values is displayed in vector and in matrix form. You should always read this output very carefully, particularly when you use your own programming statements to constrain the matrix elements. The vector form shows the mapping of the model parameters to indices of the vector $X$ that is optimized. The matrix form indicates parameter elements that are constrained by program statements by indices of $X$ in angle brackets (< >). An asterisk trailing the iteration number in the displayed optimization history of the Levenberg-Marquardt algorithm indicates that the optimization process encountered a singular Hessian matrix. When this happens, especially in the last iterations, the model may not be properly identified. The computed $\chi^2$ value of 10.337 for 7 degrees of freedom and the computed unique loadings agree with those reported by McDonald (1980), but the maximum likelihood estimates for the common factor loadings differ to some degree. The common factor loadings can be subjected to transformations that do not increase the value of the optimization criterion because the problem is not identified. An estimation problem that is not fully identified can lead to different solutions caused only by different initial values, different optimization techniques, or computers with different machine precision or floating-point arithmetic.

To overcome the identification problem in the first model, restart PROC CALIS with a simple modification to the model in which the former parameter X1 is fixed to 0. This leads to 8 instead of 7 degrees of freedom. The following code produces results that are partially displayed in Output 17.4.1.

```sas
Data ram2 (TYPE=RAM); set ram;
   if _type_ = 'ESTIM' then
      if _name_ = 'X1' then do;
         _name_ = ''; _estim_ = 0.;
```
### Example 17.4. Linear Relations Among Factor Loadings

```sas
end;
run;

proc calis data=Kinzer method=max inram=ram2 nobs=326;
  Title2 "Linearly Related Factor Analysis, (Mcdonald,1980)"
  Title3 "Identified Model"
 Parms Alfa = .5 Beta = -.5;
  X7 = Alfa;
  X8 = Alfa + Beta * X2;
  X9 = Alfa + Beta * X3;
  X10 = Alfa + Beta * X4;
  X11 = Alfa + Beta * X5;
  X12 = Alfa + Beta * X6;
  Bounds X13-X18 >= 0.;
run;
```

#### Output 17.4.1. Linearly Related Factor Analysis: Identification Problem

Data Matrix of Kinzer & Kinzer, see GUTTMAN (1957)
Linearly Related Factor Analysis, (Mcdonald,1980)
Identified Model

The CALIS Procedure
Covariance Structure Analysis: Pattern and Initial Values

<table>
<thead>
<tr>
<th>Matrix</th>
<th>Rows</th>
<th>Columns</th>
<th>----</th>
<th>Matrix Type ----</th>
</tr>
</thead>
<tbody>
<tr>
<td>Term 1</td>
<td>1 F</td>
<td>6 8</td>
<td>GENERAL</td>
<td></td>
</tr>
<tr>
<td>Term 2</td>
<td>I 8</td>
<td>8 8</td>
<td>IDENTITY</td>
<td></td>
</tr>
</tbody>
</table>

Data Matrix of Kinzer & Kinzer, see GUTTMAN (1957)
Linearly Related Factor Analysis, (Mcdonald,1980)
Identified Model

The CALIS Procedure
Covariance Structure Analysis: Maximum Likelihood Estimation

Parameter Estimates 13
Functions (Observations) 21
Lower Bounds 6
Upper Bounds 0

Optimization Start
Active Constraints 0 Objective Function 0.3234289189
Max Abs Gradient Element 2.2633860283 Radius 5.8468569273
**Output 17.4.1.** (continued)

Data Matrix of Kinzer & Kinzer, see GUTTMAN (1957)
Linearly Related Factor Analysis, (Mcdonald,1980)
Identified Model

The CALIS Procedure  
Covariance Structure Analysis: Maximum Likelihood Estimation

<table>
<thead>
<tr>
<th>Iter</th>
<th>Restarts</th>
<th>Function Calls</th>
<th>Active Constraints</th>
<th>Objective Function Change</th>
<th>Max Abs Gradient Element</th>
<th>Lambda Change</th>
<th>Ratio Between Actual and Predicted Change</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0</td>
<td>2</td>
<td>0</td>
<td>0.07994</td>
<td>0.2435</td>
<td>0.3984</td>
<td>0.557</td>
</tr>
<tr>
<td>2</td>
<td>0</td>
<td>3</td>
<td>0</td>
<td>0.03334</td>
<td>0.0466</td>
<td>0.0672</td>
<td>1.202</td>
</tr>
<tr>
<td>3</td>
<td>0</td>
<td>4</td>
<td>0</td>
<td>0.03185</td>
<td>0.00150</td>
<td>0.00439</td>
<td>1.058</td>
</tr>
<tr>
<td>4</td>
<td>0</td>
<td>5</td>
<td>0</td>
<td>0.03181</td>
<td>0.000034</td>
<td>0.00236</td>
<td>0.811</td>
</tr>
<tr>
<td>5</td>
<td>0</td>
<td>6</td>
<td>0</td>
<td>0.03181</td>
<td>3.982E-6</td>
<td>0.000775</td>
<td>0.591</td>
</tr>
<tr>
<td>6</td>
<td>0</td>
<td>7</td>
<td>0</td>
<td>0.03181</td>
<td>9.275E-7</td>
<td>0.000490</td>
<td>0.543</td>
</tr>
<tr>
<td>7</td>
<td>0</td>
<td>8</td>
<td>0</td>
<td>0.03181</td>
<td>2.402E-7</td>
<td>0.000206</td>
<td>0.526</td>
</tr>
<tr>
<td>8</td>
<td>0</td>
<td>9</td>
<td>0</td>
<td>0.03181</td>
<td>6.336E-8</td>
<td>0.000129</td>
<td>0.514</td>
</tr>
<tr>
<td>9</td>
<td>0</td>
<td>10</td>
<td>0</td>
<td>0.03181</td>
<td>1.687E-8</td>
<td>0.000054</td>
<td>0.505</td>
</tr>
<tr>
<td>10</td>
<td>0</td>
<td>11</td>
<td>0</td>
<td>0.03181</td>
<td>4.521E-9</td>
<td>0.000034</td>
<td>0.498</td>
</tr>
<tr>
<td>11</td>
<td>0</td>
<td>12</td>
<td>0</td>
<td>0.03181</td>
<td>1.217E-9</td>
<td>0.000014</td>
<td>0.493</td>
</tr>
<tr>
<td>12</td>
<td>0</td>
<td>13</td>
<td>0</td>
<td>0.03181</td>
<td>3.29E-10</td>
<td>8.971E-6</td>
<td>0.489</td>
</tr>
</tbody>
</table>

Optimization Results

<table>
<thead>
<tr>
<th>Iterations</th>
<th>Function Calls</th>
<th>12</th>
</tr>
</thead>
<tbody>
<tr>
<td>Jacobian Calls</td>
<td>13 Active Constraints</td>
<td>0</td>
</tr>
<tr>
<td>Objective Function</td>
<td>0.0318073951 Max Abs Gradient Element</td>
<td>8.9711916E-6</td>
</tr>
<tr>
<td>Lambda</td>
<td>0 Actual Over Pred Change</td>
<td>0.4888109559</td>
</tr>
<tr>
<td>Radius</td>
<td>0.0002016088</td>
<td></td>
</tr>
</tbody>
</table>

ABSGCONV convergence criterion satisfied.
Output 17.4.1. (continued)

<table>
<thead>
<tr>
<th>Fit Function</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Goodness of Fit Index (GFI)</td>
<td>0.9897</td>
</tr>
<tr>
<td>GFI Adjusted for Degrees of Freedom (AGFI)</td>
<td>0.9730</td>
</tr>
<tr>
<td>Root Mean Square Residual (RMR)</td>
<td>0.0409</td>
</tr>
<tr>
<td>Parsimonious GFI (Mulaik, 1989)</td>
<td>0.5278</td>
</tr>
<tr>
<td>Chi-Square</td>
<td>10.3374</td>
</tr>
<tr>
<td>Chi-Square DF</td>
<td>8</td>
</tr>
<tr>
<td>Pr &gt; Chi-Square</td>
<td>0.2421</td>
</tr>
<tr>
<td>Independence Model Chi-Square</td>
<td>682.87</td>
</tr>
<tr>
<td>Independence Model Chi-Square DF</td>
<td>15</td>
</tr>
<tr>
<td>RMSEA Estimate</td>
<td>0.0300</td>
</tr>
<tr>
<td>RMSEA 90% Lower Confidence Limit</td>
<td>.</td>
</tr>
<tr>
<td>RMSEA 90% Upper Confidence Limit</td>
<td>0.0756</td>
</tr>
<tr>
<td>ECVI Estimate</td>
<td>0.1136</td>
</tr>
<tr>
<td>ECVI 90% Lower Confidence Limit</td>
<td>.</td>
</tr>
<tr>
<td>ECVI 90% Upper Confidence Limit</td>
<td>0.1525</td>
</tr>
<tr>
<td>Probability of Close Fit</td>
<td>0.7137</td>
</tr>
<tr>
<td>Bentler’s Comparative Fit Index</td>
<td>0.9965</td>
</tr>
<tr>
<td>Normal Theory Reweighted LS Chi-Square</td>
<td>10.1441</td>
</tr>
<tr>
<td>Akaike’s Information Criterion</td>
<td>-5.6626</td>
</tr>
<tr>
<td>Bozdogan’s (1987) CAIC</td>
<td>-43.9578</td>
</tr>
<tr>
<td>Schwarz’s Bayesian Criterion</td>
<td>-35.9578</td>
</tr>
<tr>
<td>McDonald’s (1989) Centrality</td>
<td>0.9964</td>
</tr>
<tr>
<td>Bentler &amp; Bonett’s (1980) Non-normed Index</td>
<td>0.9934</td>
</tr>
<tr>
<td>Bentler &amp; Bonett’s (1980) NFI</td>
<td>0.9849</td>
</tr>
<tr>
<td>James, Mulaik, &amp; Brett (1982) Parsimonous NFI</td>
<td>0.5253</td>
</tr>
<tr>
<td>Z-Test of Wilson &amp; Hilferty (1931)</td>
<td>0.7019</td>
</tr>
<tr>
<td>Bollen (1986) Normed Index Rho1</td>
<td>0.9716</td>
</tr>
<tr>
<td>Bollen (1988) Non-normed Index Delta2</td>
<td>0.9965</td>
</tr>
<tr>
<td>Hoelter’s (1983) Critical N</td>
<td>489</td>
</tr>
</tbody>
</table>

Example 17.4. Linear Relations Among Factor Loadings
Output 17.4.1. (continued)

### Data Matrix of Kinzer & Kinzer, see GUTTMAN (1957)
### Linearly Related Factor Analysis, (Mcdonald,1980)
### Identified Model

#### The CALIS Procedure
#### Covariance Structure Analysis: Maximum Likelihood Estimation

**Estimated Parameter Matrix F[6:8]**

<table>
<thead>
<tr>
<th></th>
<th>Fact1</th>
<th>Fact2</th>
<th>Uvar1</th>
<th>Uvar2</th>
<th>Uvar3</th>
<th>Uvar4</th>
<th>Uvar5</th>
<th>Uvar6</th>
</tr>
</thead>
<tbody>
<tr>
<td>Obs1</td>
<td>0</td>
<td>0.7151</td>
<td>0.7283</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Obs2</td>
<td>-0.0543</td>
<td>0.7294</td>
<td>0.6707</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Obs3</td>
<td>0.1710</td>
<td>0.6703</td>
<td>0.6983</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Obs4</td>
<td>0.2922</td>
<td>0.6385</td>
<td>0</td>
<td>0</td>
<td>0.6876</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Obs5</td>
<td>0.5987</td>
<td>0.5582</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.5579</td>
<td>0</td>
</tr>
<tr>
<td>Obs6</td>
<td>0.4278</td>
<td>0.6029</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.7336</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>Fact1</th>
<th>Fact2</th>
<th>Uvar1</th>
<th>Uvar2</th>
<th>Uvar3</th>
<th>Uvar4</th>
<th>Uvar5</th>
<th>Uvar6</th>
</tr>
</thead>
<tbody>
<tr>
<td>Obs1</td>
<td>0</td>
<td>0.0405</td>
<td>0.0408</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Obs2</td>
<td>0.1042</td>
<td>0.0438</td>
<td>0.0472</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Obs3</td>
<td>0.0845</td>
<td>0.0396</td>
<td>0.0324</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>
| Obs4  | 2.0249| 16.9077| 21.5473| 0 | 0 &nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&n...
The lambda value of the iteration history indicates that Newton steps can always be performed. Because no singular Hessian matrices (which can slow down the convergence rate considerably) are computed, this example needs just 12 iterations compared to the 17 needed in the previous example. Note that the number of iterations may be machine-dependent. The value of the fit function, the residuals, and the $\chi^2$ value agree with the values obtained in fitting the first model. This indicates that this second model is better identified than the first one. It is fully identified, as indicated by the fact that the Hessian matrix is nonsingular.

Example 17.5. Ordinal Relations Among Factor Loadings

McDonald (1980) uses the same data set to compute a factor analysis with ordinally constrained factor loadings. The results of the linearly constrained factor analysis show that the loadings of the two factors are ordered as 2, 1, 3, 4, 6, 5. McDonald (1980) then tests the hypothesis that the factor loadings are all nonnegative and can be ordered in the following manner:

\begin{align*}
&b_{11} \leq b_{21} \leq b_{31} \leq b_{41} \leq b_{51} \leq b_{61} \\
&b_{12} \geq b_{22} \geq b_{32} \geq b_{42} \geq b_{52} \geq b_{62}
\end{align*}

This example is recomputed by PROC CALIS to illustrate a further application of the COSAN model statement combined with program statements. The same identification problem as in Example 17.4 on page 571 occurs here. The following model specification describes an unidentified model:

```sas
proc calis data=Kinzer method=max outram=ram tech=nr nobs=326;
Title2 "Ordinally Related Factor Analysis, (Mcdonald,1980)";
Title3 "Identification Problem";
Cosan F(8,Gen) * I(8,Ide);
MATRIX F
  [,1] = x1-x6,
  [,2] = x7-x12,
  [1,3] = x13-x18;
PARAMETERS t1-t10=1.;
x2 = x1 + t1 * t1;
x3 = x2 + t2 * t2;
x4 = x3 + t3 * t3;
x5 = x4 + t4 * t4;
x6 = x5 + t5 * t5;
x11 = x12 + t6 * t6;
x10 = x11 + t7 * t7;
x9 = x10 + t8 * t8;
x8 = x9 + t9 * t9;
x7 = x8 + t10 * t10;
Bounds x13-x18 >= 0.;
Vnames F Fact1 Fact2 Uvar1-Uvar6;
run;
```

You can specify the same model with the LINCON statement:
To have an identified model, the loading, $b_{11} (x1)$, is fixed at 0. The information in the OUTRAM= data set (the data set ram), produced by the unidentified model, can be used to specify the identified model. However, because $x1$ is now a fixed constant in the identified model, it should not have a parameter name in the new analysis. Thus, the data set ram is modified as follows:

```sas
data ram2(TYPE=RAM); set ram;
  if _name_ = 'x1' then do;
    _name_ = ' '; _estim_ = 0.;
  end;
run;
```

The data set ram2 is now an OUTRAM= data set in which $x1$ is no longer a parameter. PROC CALIS reads the information (that is, the set of parameters and the model specification) in the data set ram2 for the identified model. As displayed in the following code, you can use the PARMS statement to specify the desired ordinal relationships between the parameters.

```sas
proc calis data=Kinzer method=max inram=ram2 tech=nr nobs=326;
  title2 "Ordinally Related Factor Analysis, (McDonald,1980)";
  title3 "Identified Model with X1=0";
  parms t1-t10= 10 * 1.;
  x2 = + t1 * t1;
  x3 = x2 + t2 * t2;
  x4 = x3 + t3 * t3;
  x5 = x4 + t4 * t4;
  x6 = x5 + t5 * t5;
  x11 = x12 + t6 * t6;
  x10 = x11 + t7 * t7;
  x9 = x10 + t8 * t8;
```
Example 17.5. Ordinal Relations Among Factor Loadings

\[ x_8 = x_9 + t_9 \times t_9; \]
\[ x_7 = x_8 + t_{10} \times t_{10}; \]
Bounds \( x_{13} - x_{18} \geq 0 \);
run;

Selected output for the identified model is displayed in Output 17.5.1.

Output 17.5.1. Factor Analysis with Ordinal Constraints

Data Matrix of Kinzer & Kinzer, see Guttman (1957)
Ordinarily Related Factor Analysis, (Mcdonald,1980)
Identified Model with \( X_1=0 \)

The CALIS Procedure
Covariance Structure Analysis: Maximum Likelihood Estimation

Parameter Estimates 17
Functions (Observations) 21
Lower Bounds 6
Upper Bounds 0

Optimization Start
Active Constraints 0 Objective Function 5.2552270182
Max Abs Gradient Element 0.8821788922

<table>
<thead>
<tr>
<th>Iter</th>
<th>Restarts</th>
<th>Function Calls</th>
<th>Active Constraints</th>
<th>Objective Function</th>
<th>Objective Function Change</th>
<th>Max Abs Gradient Element</th>
<th>Ridge</th>
<th>Predicted Change</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0</td>
<td>2</td>
<td>0</td>
<td>3.14901</td>
<td>2.1062</td>
<td>1.0712</td>
<td>0</td>
<td>2.226</td>
</tr>
<tr>
<td>2</td>
<td>0</td>
<td>3</td>
<td>0</td>
<td>1.42725</td>
<td>1.7218</td>
<td>1.0902</td>
<td>0</td>
<td>2.064</td>
</tr>
<tr>
<td>3</td>
<td>0</td>
<td>4</td>
<td>0</td>
<td>0.41661</td>
<td>1.0106</td>
<td>0.7472</td>
<td>0</td>
<td>1.731</td>
</tr>
<tr>
<td>4</td>
<td>0</td>
<td>5</td>
<td>0</td>
<td>0.09260</td>
<td>0.3240</td>
<td>0.3365</td>
<td>0</td>
<td>1.314</td>
</tr>
<tr>
<td>5</td>
<td>0</td>
<td>6</td>
<td>0</td>
<td>0.09186</td>
<td>0.00731</td>
<td>0.3880</td>
<td>0</td>
<td>0.0123</td>
</tr>
<tr>
<td>6</td>
<td>0</td>
<td>8</td>
<td>0</td>
<td>0.04570</td>
<td>0.0462</td>
<td>0.2870</td>
<td>0</td>
<td>0.0313</td>
</tr>
<tr>
<td>7</td>
<td>0</td>
<td>10</td>
<td>0</td>
<td>0.03269</td>
<td>0.0130</td>
<td>0.0909</td>
<td>0</td>
<td>0.0031</td>
</tr>
<tr>
<td>8</td>
<td>0</td>
<td>16</td>
<td>0</td>
<td>0.02771</td>
<td>0.00498</td>
<td>0.0890</td>
<td>0</td>
<td>0.0080</td>
</tr>
<tr>
<td>9</td>
<td>0</td>
<td>17</td>
<td>0</td>
<td>0.02602</td>
<td>0.00168</td>
<td>0.0174</td>
<td>0</td>
<td>0.0040</td>
</tr>
<tr>
<td>10</td>
<td>0</td>
<td>19</td>
<td>0</td>
<td>0.02570</td>
<td>0.00323</td>
<td>0.0141</td>
<td>0</td>
<td>0.0080</td>
</tr>
<tr>
<td>11</td>
<td>0</td>
<td>21</td>
<td>0</td>
<td>0.02560</td>
<td>0.000103</td>
<td>0.00179</td>
<td>0</td>
<td>1.170</td>
</tr>
<tr>
<td>12</td>
<td>0</td>
<td>23</td>
<td>0</td>
<td>0.02559</td>
<td>7.587E-6</td>
<td>0.000670</td>
<td>0.160</td>
<td>1.423</td>
</tr>
<tr>
<td>13</td>
<td>0</td>
<td>24</td>
<td>0</td>
<td>0.02559</td>
<td>2.993E-6</td>
<td>0.000402</td>
<td>0.160</td>
<td>1.010</td>
</tr>
<tr>
<td>14</td>
<td>0</td>
<td>27</td>
<td>0</td>
<td>0.02559</td>
<td>1.013E-6</td>
<td>0.000206</td>
<td>0.160</td>
<td>1.388</td>
</tr>
<tr>
<td>15</td>
<td>0</td>
<td>28</td>
<td>0</td>
<td>0.02559</td>
<td>1.889E-7</td>
<td>0.000202</td>
<td>0.0400</td>
<td>0.530</td>
</tr>
<tr>
<td>16</td>
<td>0</td>
<td>30</td>
<td>0</td>
<td>0.02559</td>
<td>1.803E-7</td>
<td>0.000097</td>
<td>0.0800</td>
<td>0.630</td>
</tr>
<tr>
<td>17</td>
<td>0</td>
<td>32</td>
<td>0</td>
<td>0.02559</td>
<td>4.845E-8</td>
<td>0.000035</td>
<td>0.160</td>
<td>1.340</td>
</tr>
<tr>
<td>18</td>
<td>0</td>
<td>33</td>
<td>0</td>
<td>0.02559</td>
<td>1.837E-9</td>
<td>0.000049</td>
<td>0.0400</td>
<td>0.125</td>
</tr>
<tr>
<td>19</td>
<td>0</td>
<td>35</td>
<td>0</td>
<td>0.02559</td>
<td>9.39E-9</td>
<td>0.000024</td>
<td>0.0800</td>
<td>0.579</td>
</tr>
<tr>
<td>20</td>
<td>0</td>
<td>37</td>
<td>0</td>
<td>0.02559</td>
<td>2.552E-9</td>
<td>6.176E-6</td>
<td>0.160</td>
<td>1.305</td>
</tr>
</tbody>
</table>

Optimization Results
Iterations 20  Function Calls 38
Jacobian Calls 21  Active Constraints 0
Objective Function 0.0255871615  Max Abs Gradient Element 6.1764582E-6
Ridge 0.04  Actual Over Pred Change 1.3054374955

ABSGCONV convergence criterion satisfied.
### Output 17.5.1.  (continued)

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Data Matrix of Kinzer &amp; Kinzer, see GUTTMAN (1957)</td>
<td></td>
</tr>
<tr>
<td>Ordinally Related Factor Analysis, (Mcdonald,1980)</td>
<td></td>
</tr>
<tr>
<td>Identified Model with X1=0</td>
<td></td>
</tr>
<tr>
<td>The CALIS Procedure</td>
<td></td>
</tr>
<tr>
<td>Covariance Structure Analysis: Maximum Likelihood Estimation</td>
<td></td>
</tr>
<tr>
<td>Fit Function</td>
<td>0.0256</td>
</tr>
<tr>
<td>Goodness of Fit Index (GFI)</td>
<td>0.9916</td>
</tr>
<tr>
<td>GFI Adjusted for Degrees of Freedom (AGFI)</td>
<td>0.9557</td>
</tr>
<tr>
<td>Root Mean Square Residual (RMR)</td>
<td>0.0180</td>
</tr>
<tr>
<td>Parsimonious GFI (Mulaik, 1989)</td>
<td>0.2644</td>
</tr>
<tr>
<td>Chi-Square</td>
<td>8.3158</td>
</tr>
<tr>
<td>Chi-Square DF</td>
<td>4</td>
</tr>
<tr>
<td>Pr &gt; Chi-Square</td>
<td>0.0807</td>
</tr>
<tr>
<td>Independence Model Chi-Square</td>
<td>682.87</td>
</tr>
<tr>
<td>Independence Model Chi-Square DF</td>
<td>15</td>
</tr>
<tr>
<td>RMSEA Estimate</td>
<td>0.0576</td>
</tr>
<tr>
<td>RMSEA 90% Lower Confidence Limit</td>
<td></td>
</tr>
<tr>
<td>RMSEA 90% Upper Confidence Limit</td>
<td>0.1133</td>
</tr>
<tr>
<td>ECVI Estimate</td>
<td>0.1325</td>
</tr>
<tr>
<td>ECVI 90% Lower Confidence Limit</td>
<td></td>
</tr>
<tr>
<td>ECVI 90% Upper Confidence Limit</td>
<td>0.1711</td>
</tr>
<tr>
<td>Probability of Close Fit</td>
<td>0.3399</td>
</tr>
<tr>
<td>Bentler’s Comparative Fit Index</td>
<td>0.9935</td>
</tr>
<tr>
<td>Normal Theory Reweighted LS Chi-Square</td>
<td>8.2901</td>
</tr>
<tr>
<td>Akaike’s Information Criterion</td>
<td>0.3158</td>
</tr>
<tr>
<td>Bozdogan’s (1987) CAIC</td>
<td>-18.8310</td>
</tr>
<tr>
<td>Schwarz’s Bayesian Criterion</td>
<td>-14.8310</td>
</tr>
<tr>
<td>McDonald’s (1989) Centrality</td>
<td>0.9934</td>
</tr>
<tr>
<td>Bentler &amp; Bonett’s (1980) Non-normed Index</td>
<td>0.9758</td>
</tr>
<tr>
<td>Bentler &amp; Bonett’s (1980) NFI</td>
<td>0.9878</td>
</tr>
<tr>
<td>James, Mulaik, &amp; Brett (1982) Parsimonious NFI</td>
<td>0.2634</td>
</tr>
<tr>
<td>Z-Test of Wilson &amp; Hilferty (1931)</td>
<td>1.4079</td>
</tr>
<tr>
<td>Bollen (1986) Normed Index Rhol</td>
<td>0.9543</td>
</tr>
<tr>
<td>Bollen (1988) Non-normed Index Delta2</td>
<td>0.9936</td>
</tr>
<tr>
<td>Hoelter’s (1983) Critical N</td>
<td>372</td>
</tr>
</tbody>
</table>
Output 17.5.1.  (continued)

Data Matrix of Kinzer & Kinzer, see GUTTMAN (1957)
Ordinally Related Factor Analysis, (Mcdonald,1980)
Identified Model with X1=0

The CALIS Procedure
Covariance Structure Analysis: Maximum Likelihood Estimation

Estimated Parameter Matrix F[6:8]
Standard Errors and t Values
General Matrix

<table>
<thead>
<tr>
<th></th>
<th>Fact1</th>
<th>Fact2</th>
<th>Uvar1</th>
<th>Uvar2</th>
<th>Uvar3</th>
<th>Uvar4</th>
<th>Uvar5</th>
<th>Uvar6</th>
</tr>
</thead>
<tbody>
<tr>
<td>Obs1</td>
<td>0</td>
<td>0.7101</td>
<td>0.7131</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>0</td>
<td>0.0435</td>
<td>0.0404</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>0</td>
<td>16.3317</td>
<td>17.6427</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Obs2</td>
<td>0.0261</td>
<td>0.7101</td>
<td>0</td>
<td>0.6950</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>0.0875</td>
<td>0.0435</td>
<td>0</td>
<td>0.0391</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>0.2977</td>
<td>16.3317</td>
<td>0</td>
<td>17.7571</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Obs3</td>
<td>0.2382</td>
<td>0.6827</td>
<td>0</td>
<td>0</td>
<td>0.6907</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>0.0851</td>
<td>0.0604</td>
<td>0</td>
<td>0</td>
<td>0.0338</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>2.7998</td>
<td>11.3110</td>
<td>0</td>
<td>0</td>
<td>20.4239</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Obs4</td>
<td>0.3252</td>
<td>0.6580</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.6790</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>0.0823</td>
<td>0.0621</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.0331</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>3.9504</td>
<td>10.5950</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>20.5361</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Obs5</td>
<td>0.5395</td>
<td>0.5528</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.6249</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>0.0901</td>
<td>0.0705</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.0534</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>5.9887</td>
<td>7.8359</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>11.7052</td>
<td>0</td>
</tr>
<tr>
<td>Obs6</td>
<td>0.5395</td>
<td>0.4834</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.7005</td>
</tr>
<tr>
<td></td>
<td>0.0918</td>
<td>0.0726</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.0524</td>
</tr>
<tr>
<td></td>
<td>5.8776</td>
<td>6.6560</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>13.3749</td>
</tr>
</tbody>
</table>
Output 17.5.1. (continued)

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Estimate</th>
<th>Standard Error</th>
<th>t Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>t1</td>
<td>0.16143</td>
<td>0.27111</td>
<td>0.60</td>
</tr>
<tr>
<td>t2</td>
<td>0.46060</td>
<td>0.09289</td>
<td>4.96</td>
</tr>
<tr>
<td>t3</td>
<td>0.29496</td>
<td>0.13702</td>
<td>2.15</td>
</tr>
<tr>
<td>t4</td>
<td>0.46297</td>
<td>0.10756</td>
<td>4.30</td>
</tr>
<tr>
<td>t5</td>
<td>0.0000522</td>
<td>1311</td>
<td>0.00</td>
</tr>
<tr>
<td>t6</td>
<td>0.26347</td>
<td>0.12203</td>
<td>2.16</td>
</tr>
<tr>
<td>t7</td>
<td>0.32430</td>
<td>0.09965</td>
<td>3.25</td>
</tr>
<tr>
<td>t8</td>
<td>0.15721</td>
<td>0.21134</td>
<td>0.74</td>
</tr>
<tr>
<td>t9</td>
<td>0.16543</td>
<td>0.20537</td>
<td>0.81</td>
</tr>
<tr>
<td>t10</td>
<td>-4.2528E-7</td>
<td>0.47736</td>
<td>-0.00</td>
</tr>
<tr>
<td>x7</td>
<td>0.71007</td>
<td>0.04348</td>
<td>16.33</td>
</tr>
<tr>
<td>x2</td>
<td>0.02606</td>
<td>0.08753</td>
<td>0.30</td>
</tr>
<tr>
<td>x8</td>
<td>0.71007</td>
<td>0.04348</td>
<td>16.33</td>
</tr>
<tr>
<td>x3</td>
<td>0.23821</td>
<td>0.08508</td>
<td>2.80</td>
</tr>
<tr>
<td>x9</td>
<td>0.68270</td>
<td>0.06036</td>
<td>11.31</td>
</tr>
<tr>
<td>x4</td>
<td>0.32521</td>
<td>0.08232</td>
<td>3.95</td>
</tr>
<tr>
<td>x10</td>
<td>0.65799</td>
<td>0.06210</td>
<td>10.60</td>
</tr>
<tr>
<td>x5</td>
<td>0.53955</td>
<td>0.09009</td>
<td>5.99</td>
</tr>
<tr>
<td>x11</td>
<td>0.55282</td>
<td>0.07055</td>
<td>7.84</td>
</tr>
<tr>
<td>x6</td>
<td>0.53955</td>
<td>0.09180</td>
<td>5.88</td>
</tr>
</tbody>
</table>

By fixing the loading \( b_{11} (x1) \) to constant 0, you obtain \( \chi^2 = 8.316 \) on \( df = 4 \) \( (p < .09) \). McDonald reports the same \( \chi^2 \) value, but on \( df = 3 \), and thus, he obtains a smaller \( p \)-value. An analysis without the fixed loading shows typical signs of an unidentified problem: after more iterations it leads to a parameter set with a \( \chi^2 \) value of 8.174 on \( df = 3 \). A singular Hessian matrix occurs.

The singular Hessian matrix of the unidentified problem slows down the convergence rate of the Levenberg-Marquardt algorithm considerably. Compared to the unidentified problem with 30 iterations, the identified problem needs only 20 iterations. Note that the number of iterations may depend on the precision of the processor.

The same model can also be specified using the LINCON statement for linear constraints:

```
proc calis DATA=Kinzer METHOD=MAX TECH=LM EDF=325;
  Title3 "Identified Model 2";
  COSAN F(8,GEN)*I(8,IDE);
  MATRIX F
      [,1] = 0.  x2-x6,
      [,2] = x7-x12,
      [1,3] = x13-x18;
```
Example 17.6. Longitudinal Factor Analysis

The following example (McDonald 1980) illustrates both the ability of PROC CALIS to formulate complex covariance structure analysis problems by the generalized COSAN matrix model and the use of program statements to impose nonlinear constraints on the parameters. The example is a longitudinal factor analysis using the Swaminathan (1974) model. For \( m = 3 \) tests, \( k = 3 \) occasions, and \( r = 2 \) factors the matrix model is formulated in the section “First-Order Autoregressive Longitudinal Factor Model” on page 402 as follows:

\[
C = F_1 F_2 F_3 LF_3^{-1} F_2^{-1} P (F_2^{-1})' (F_3^{-1})' + U^2
\]

\[
F_1 = \begin{pmatrix}
B_1 \\
B_2 \\
B_3
\end{pmatrix}, \quad F_2 = \begin{pmatrix}
I_2 & D_2 \\
D_2 & D_3
\end{pmatrix}, \quad F_3 = \begin{pmatrix}
I_2 \\
I_2
\end{pmatrix}
\]

\[
L = \begin{pmatrix}
I_2 & I_2 & I_2 \\
I_2 & I_2 & I_2 \\
I_2 & I_2 & I_2
\end{pmatrix}, \quad P = \begin{pmatrix}
I_2 & S_2 \\
S_2 & S_3
\end{pmatrix}, \quad U = \begin{pmatrix}
U_{11} & U_{12} & U_{13} \\
U_{21} & U_{22} & U_{23} \\
U_{31} & U_{32} & U_{33}
\end{pmatrix}
\]

\[
S_2 = I_2 - D_2^2, \quad S_3 = I_2 - D_3^2
\]

The Swaminathan longitudinal factor model assumes that the factor scores for each \( (m) \) common factor change from occasion to occasion \( (k) \) according to a first-order autoregressive scheme. The matrix \( F_1 \) contains the \( k \) factor loading matrices \( B_1, B_2, B_3 \) (each is \( n \times m \)). The matrices \( D_2, D_3, S_2, S_3 \) and \( U_{ij}, i, j = 1, \ldots, k \), are diagonal, and the matrices \( D_i \) and \( S_i, i = 2, \ldots, k \), are subjected to the constraint

\[
S_i + D_i^2 = I
\]

Since the constructed correlation matrix given in McDonald’s (1980) paper is singular, only unweighted least-squares estimates can be computed.
Chapter 17. The CALIS Procedure

datalines;
Obs1  1.000  .  .  .  .  .  .  .  .  .
Obs2  .100  1.000  .  .  .  .  .  .
Obs3  .250  .400  1.000  .  .  .  .  .
Obs4  .720  .108  .270  1.000  .  .  .  .
Obs5  .135  .740  .380  .180  1.000  .  .  .
Obs6  .270  .318  .800  .360  .530  1.000  .  .  .
Obs7  .650  .054  .135  .730  .090  .180  1.000  .  .  .
Obs8  .108  .690  .196  .144  .700  .269  .200  1.000  .  .
;
proc calis data=Mcdon method=ls tech=nr nobs=100;
Cosan B(6,Gen) * D1(6,Dia) * D2(6,Dia) * T(6,Low) * D3(6,Dia,Inv) *
D4(6,Dia,Inv) * P(6,Dia) + U(9,Sym);

Matrix B
[ 1,1] = X1-X3,
[ 1,2] = 0. X4-X5,
[ 1,3] = 3 * 0. X6-X8,
[ 1,4] = 4 * 0. X9-X10,
[ 1,5] = 6 * 0. X11-X13,
[ 1,6] = 7 * 0. X14-X15;

Matrix D1
[ 1,1] = 2 * 1. X16 X17 X16 X17;

Matrix D2
[ 1,1] = 4 * 1. X18 X19;

Matrix T
[ 1,1] = 6 * 1. ,
[ 3,1] = 4 * 1. ,
[ 5,1] = 2 * 1. ;

Matrix D3
[ 1,1] = 4 * 1. X18 X19;

Matrix D4
[ 1,1] = 2 * 1. X16 X17 X16 X17;

Matrix P
[ 1,1] = 2 * 1. X20-X23;

Matrix U
[ 1,1] = X24-X32,
[ 4,1] = X33-X38,
[ 7,1] = X39-X41;

Bounds 0. <= X24-X32,
-1. <= X16-X19 <= 1. ;
X20 = 1. - X16 * X16;
X21 = 1. - X17 * X17;
X22 = 1. - X18 * X18;
X23 = 1. - X19 * X19;
run;

Because this formulation of Swaminathan’s model in general leads to an unidentified problem, the results given here are different from those reported by McDonald (1980). The displayed output of PROC CALIS also indicates that the fitted central model matrices \( P \) and \( U \) are not positive definite. The BOUNDS statement constrains the diagonals of the matrices \( P \) and \( U \) to be nonnegative, but this cannot prevent \( U \)
from having three negative eigenvalues. The fact that many of the published results for more complex models in covariance structure analysis are connected to unidentified problems implies that more theoretical work should be done to study the general features of such models.

References


